

Implementasi Sistem Operasi Real-Time pada Arduino Nano dengan media Komunikasi NRF24L01 Untuk Pengukuran Suhu, Kelembaban, dan Intensitas Cahaya

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Eka Nanda Sugianto
NIM: 135150301111139



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

Implementasi Sistem Operasi Real-Time pada Arduino Nano dengan media Komunikasi NRF24L01 Untuk Pengukuran Suhu, Kelembaban, dan Intensitas Cahaya

PENGESAHAN

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Eka Nanda Sugianto

NIM: 135150301111139

Skripsi ini telah diuji dan dinyatakan lulus pada
27 Desember 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Wiyaya Kurniawan, S.T, M.T

NIP: 19820125 201504 1 002

Dosen Pembimbing II



Dahnil Syaedy, S.T., M.T., M.Sc.

NIK: 2016078704231002

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP. 197105182003121001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 27 Desember 2018



Eka Nanda Sugianto

NIM: 135150301111139



KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadiran Tuhan Yesus Kristus, oleh karena anugerah-Nya yang melimpah, kemurahan dan kasih setia yang besar akhirnya penulis dapat menyelesaikan skripsi dengan judul “Implementasi Sistem Operasi Real-Time pada Arduino Nano dengan media Komunikasi NRF24L01 Untuk Pengukuran Suhu, Kelembaban, dan Intensitas Cahaya” ini.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh sebab itu, penulis menyampaikan rasa hormat dan terima kasih kepada :

1. Tuhan yang Maha Esa karena telah menyertai dan memberkati saya pada setiap proses di dalam penyelesaian skripsi ini.
2. Tante Patty Erni Winata, Alm. Tante Yenny Winata dan Ibu Tini Winata serta keluarga yang telah membantu saya di dalam doa dan segala dukungan yang telah diberikan.
3. Pdt. Hartinez Lie S.Th dan Ev. Yafet E Wahyudi S.Th yang telah mendukung di dalam doa, solusi dalam menyelesaikan masalah dan segala dukungan yang telah diberikan.
4. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
5. Bapak Heru Nurwasito, Ir., M.Kom. selaku Wakil Ketua I Bidang Akademik Fakultas Ilmu Komputer Universitas Brawijaya Malang.
6. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya.
7. Bapak Muhammad Tanzil Furqon, S.Kom, M.CompSc selaku Sekretaris Jurusan Teknik Informatika Universitas Brawijaya.
8. Bapak Wijaya Kurniawan, S.T, M.T. selaku dosen pembimbing I yang telah memberikan bimbingan kepada penulis sehingga penulis dapat menyelesaikan skripsi ini.
9. Bapak Dahnia Syauqy, S.T, M.T., M.Sc. selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang dan dosen pembimbing II yang telah memberikan pengarahan dalam menyelesaikan laporan skripsi.
10. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng. selaku dosen pembimbing I semester sebelumnya karena kesabaran dan arahan yang telah diberikan.
11. Seluruh dosen dan karyawan Fakultas Ilmu Komputer yang telah berperan dalam proses penyelesaian skripsi ini.
12. Teman-teman kontrakan “Goyang Piranha” yaitu Zainul Bahar, Sutikno, Riko Andianto, Dimas Nofian, Tio Rendi, dan Edi atas dukungan semangat dan segala bentuk bantuan yang telah diberikan.

13. Teman-teman grup Line “Ceria” yaitu Bobby Naghi, Zonnete Bryllian, Rivaldi Wibowo, Reza Hastuti, Zhafran Wadiansyah, Hafizhuddin dan Raden Galih atas waktu dan kebersamaan selama bergaul menjadi mahasiswa.
14. Yohanes Yance sebagai teman konseling yang siap untuk selalu mendengarkan keluhan penulis dan memberikan solusi serta dukungan.

Saya menyadari bahwa laporan skripsi ini jauh dari sempurna, oleh karena itu untuk segala kritik dan saran yang membangun saya ucapkan terimakasih. Saya mengharapkan semoga laporan skripsi ini dapat berguna untuk saya dan teman-teman yang ingin mengembangkan penelitian sebagai langkah penyempurnaan sistem.

Malang, 12 Desember 2018

Penulis

ekananda007@gmail.com



ABSTRAK

Dalam membangun sistem sensor node pada monitoring rumah cerdas diperlukan adanya sebuah sistem operasi supaya task-task yang ada dapat dieksekusi secara simultan. Dengan berjalannya task-task secara simultan maka kinerja dari sistem monitoring rumah cerdas diharapkan dapat berjalan dengan semestinya. Jika tidak, maka sistem monitoring rumah cerdas akan bekerja tidak semestinya seperti mengalami penurunan atau kenaikan suhu yang secara signifikan dikarenakan kurang tepat waktu dalam memproses data serta terlambatnya pengambilan sebuah keputusan yang seharusnya tidak terjadi pada sistem monitoring rumah cerdas. Selain itu, sistem operasi yang digunakan haruslah bersifat real-time karena sebuah sistem monitoring rumah cerdas diharapkan dapat membuat keputusan dan memberikan informasi yang tepat dalam waktu yang aktual. Untuk mengatasi permasalahan di atas, RTOS (Real Time Operating System) adalah salah satu sistem operasi yang tepat dalam mengeksekusi task-task yang ada secara simultan dan bersifat real-time. Untuk menerapkan RTOS pada mikrokontroler Arduino Nano dibutuhkan library FreeRTOS. Sistem sensor node ini dibagi menjadi 2 yaitu node client dan node base. Media komunikasi wireless yang digunakan yaitu nRF24L01. Node client terdiri atas mikrokontroler Arduino Nano, sensor LDR, sensor DHT11, dan nRF24L01. Sedangkan pada node base terdiri atas mikrokontroler Arduino Nano dan nRF24L01. Jumlah task pada node client adalah 3 sedangkan pada node base adalah 2. Metode yang digunakan adalah Preemptive Priority Based-Scheduling. Dari hasil pengujian, task-task yang ada dapat berjalan sesuai dengan prioritas yang diberikan. Fungsi vTaskDelay untuk mengatur waktu eksekusi setiap task telah berjalan sebagaimana mestinya.

Kata kunci: RTOS, FreeRTOS, Sensor Node, Monitoring Rumah Cerdas, Simultan, Prioritas.

ABSTRACT

In building system node sensor for smart home monitoring needed an operating system so that tasks can be executed simultaneously. By running tasks simultaneously, performance of the system monitoring smart home expected to running properly. If not, then the smart home monitoring system will work improperly such as experiencing a decrease or increase in temperature significantly due to lack of timely processing of data and the delay in making a decision that should not occur in a smart home monitoring system. In addition, the operating system must be in real-time because a smart home monitoring system is expected to be able to make decisions and provide the right information in the actual time. To resolve the above problems, RTOS (Real Time Operating System) is one of the appropriate operating systems in executing tasks simultaneously and it's also real-time. To apply RTOS to the Arduino Nano microcontroller, FreeRTOS library is needed. This sensor node system is divided into 2 node, namely client node and base node. The wireless communication media is using nRF24L01. The client node consists of the Arduino Nano microcontroller, LDR sensor, DHT11 sensor, and nRF24L01. While the base node consists of an Arduino Nano microcontroller and nRF24L01. The number of tasks on the client node is 3 tasks while at the node base is 2 tasks. Implementation for method is using Preemptive Priority Based-Scheduling. From the test results, tasks can run according to the priority given. The `vTaskDelay` function to set the execution time of each task is running as it should.

Keyword: RTOS, FreeRTOS, Node Sensor, Smart Home Monitoring, simultaneously, priority.

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	ii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	2
1.2 Rumusan masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	4
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Tinjauan Pustaka	6
2.2 Dasar Teori.....	7
2.2.1 Sensor Pengukuran Intensitas Cahaya.....	7
2.2.2 Sensor Pengukur Suhu dan Kelembaban	8
2.2.3 Mikrokontroler Arduino Nano	9
2.2.4 Arduino IDE	11
2.2.5 Modul <i>Wireless</i> NRF24L01	13
2.2.6 Lampu LED.....	13
2.2.7 Kipas Motor DC	14
2.2.8 Sistem Operasi Real-Time	15
2.2.9 <i>Free</i> Sistem Operasi <i>Real-Time</i>	16
BAB 3 METODOLOGI	17
3.1 Studi Literatur	18
3.2 Analisis Kebutuhan	19
3.2.1 Kebutuhan Perangkat Keras.....	19

3.2.2 Kebutuhan Perangkat Lunak	20
3.3 Gambaran Umum Perancangan	20
3.4 Implementasi	21
3.5 Pengujian Sistem.....	21
3.6 Analisis dan Pembahasan	21
3.7 Kesimpulan.....	22
3.8 Pembuatan Laporan Skripsi	22
BAB 4 REKAYASA KEBUTUHAN.....	22
4.1 Gambaran Umum Sistem.....	23
4.2 Kebutuhan Sistem.....	23
4.2.1 Kebutuhan Fungsional.....	23
4.2.2 Kebutuhan Non Fungsional.....	24
4.2.3 Kebutuhan Perangkat Keras.....	25
4.2.4 Kebutuhan Perangkat Lunak.....	26
4.3 Batasan Desain Sistem.....	26
BAB 5 PERANCANGAN DAN IMPLEMENTASI	28
5.1 Perancangan Sistem.....	28
5.1.1 Pendefinisian dan Perancangan Penelitian.....	28
5.1.2 Perancangan <i>Hardware</i>	28
5.1.3 Perancangan <i>Task</i> pengambilan data sensor LDR pada Node <i>Client</i>	35
5.1.4 Perancangan <i>Task</i> pengambilan data sensor DHT11 pada Node <i>Client</i>	37
5.1.5 Perancangan <i>Task</i> Pengiriman Data pada Node <i>Client</i>	38
5.1.6 Perancangan Metode <i>Preemptive Priority-Based Scheduling</i> pada Node <i>Client</i>	35
5.1.7 Perancangan <i>Task</i> Penerimaan Data pada Node <i>Base</i>	40
5.1.8 Perancangan <i>Task</i> pencetakan data sensor yang telah diterima pada Node <i>Base</i>	41
5.1.9 Perancangan Metode <i>Preemptive Priority-Based Scheduling</i> pada Node <i>Base</i>	40
5.2 Implementasi	42
5.2.1 Spesifikasi <i>Hardware</i>	42
5.2.2 Implementasi Komunikasi <i>Hardware</i>	43

5.2.3 Implementasi <i>Hardware</i>	44
5.2.4 Implementasi <i>Software</i>	45
BAB 6 PeNGUJIAN DAN ANALISIS.....	53
6.1 Pengujian Sensor DHT11.....	53
6.1.1 Tujuan.....	53
6.1.2 Prosedur	53
6.1.3 Hasil dan Analisis.....	54
6.2 Pengujian Media Komunikasi Wireless nRF24L01	55
6.2.1 Tujuan.....	55
6.2.2 Prosedur	55
6.2.3 Hasil dan Analisis.....	56
6.3 Pengujian Metode <i>Preemptive Priority Based-Scheduling</i>	57
6.3.1 Tujuan.....	57
6.3.2 Prosedur.....	57
6.3.3 Hasil dan Analisis.....	57
BAB 7 PENUTUP	59
7.1 Kesimpulan.....	59
7.2 Saran	59
Bibliography	Error! Bookmark not defined.

DAFTAR GAMBAR

Gambar 2. 1 Sensor LDR.....	7
Gambar 2. 2 Sensor DHT11	8
Gambar 2. 3 Arduino Nano	10
Gambar 2. 4 Tampilan <i>software</i> Arduino IDE	12
Gambar 2. 5 Modul <i>wireless</i> nRF24L01.....	13
Gambar 2. 6 Lampu LED.....	13
Gambar 2. 7 Kipas Motor DC.....	14
Gambar 3. 1 Proses metode penelitian	18
Gambar 3. 2 Perancangan Sistem.....	20
Gambar 5. 1 Diagram blok node <i>base</i>	29
Gambar 5. 2 Perancangan sistem node <i>base</i>	29
Gambar 5. 3 Diagram blok node <i>client</i>	31
Gambar 5.4 Perancangan Sistem Node <i>Client</i>	32
Gambar 5. 5 Diagram alur metode <i>Preemptive Priority-Based Scheduling</i> pada Node <i>Client</i>	36
Gambar 5. 6 Diagram alur pengambilan data sensor LDR.....	37
Gambar 5. 7 Diagram alur pengambilan data sensor DHT11	38
Gambar 5. 8 Diagram alur pengiriman data sensor pada node <i>Client</i>	39
Gambar 5. 9 Diagram alur metode <i>preemptive priority based-scheduling</i> pada node <i>base</i>	40
Gambar 5. 10 Diagram alur penerimaan data pada node <i>base</i>	41
Gambar 5. 11 Diagram alur pencetakan data sensor pada node <i>base</i>	42
Gambar 5.12 Implementasi nRF24L01 sebagai Komunikasi <i>Wireless</i>	44
Gambar 5.13 Komponen <i>Hardware</i>	44
Gambar 6. 1 Grafik pengujian dan analisis	53
Gambar 6. 2 Analisis hasil pengujian metode <i>Preemptive Priority Based-Scheduling</i> berbentuk Grafik Garis.....	58

DAFTAR TABEL

Tabel 2. 1 Tabel Perbandingan Penelitian.....	6
Tabel 2. 2 Spesifikasi Sensor LDR	8
Tabel 2. 3 Spesifikasi Sensor DHT11.....	9
Tabel 5. 1 Konfigurasi pin Arduino Nano dengan modul komunikasi <i>wireless nRF24L01</i> pada node <i>base</i>	29
Tabel 5. 2 Konfigurasi fungsi dan prioritas <i>task</i> pada node <i>base</i>	30
Tabel 5. 3 Konfigurasi pin sensor DHT11 dengan mikrokontroler Arduino Nano pada node <i>client</i>	32
Tabel 5. 4 Konfigurasi pin mikrokontroler Arduino Nano dengan sensor LDR pada node <i>client</i>	33
Tabel 5.5 Konfigurasi Pin mikrokontroler Arduino Nano dengan modul komunikasi <i>wireless nRF24L01</i> pada node <i>client</i>	33
Tabel 5. 6 Konfigurasi Pin Mikrokontroler Arduino Nano dengan kipas	33
Tabel 5. 7 Konfigurasi pin mikrokontroler Arduino Nano dengan LED.....	34
Tabel 5. 8 Konfigurasi prioritas dan fungsi <i>task</i> pada node <i>client</i>	35
Tabel 5. 9 Spesifikasi Laptop	43
Tabel 5. 10 <i>Library code</i> node <i>client</i>	45
Tabel 5. 11 <i>Library code</i> node <i>base</i>	45
Tabel 5. 12 Kode program inialisasi <i>task</i> pada node <i>client</i>	46
Tabel 5. 13 Kode program inialisasi <i>task</i> pada node <i>base</i>	46
Tabel 5. 14 Kode program pemberian prioritas pada node <i>client</i>	46
Tabel 5. 15 Kode program pemberian prioritas pada node <i>base</i>	47
Tabel 5. 16 Kode program <i>task</i> sensor LDR pada node <i>client</i>	48
Tabel 5. 17 Kode program <i>task</i> sensor DHT11 pada node <i>client</i>	49
Tabel 5. 18 Kode program <i>task</i> pengiriman data sensor pada node <i>client</i>	50
Tabel 5. 19 Kode program <i>task</i> penerimaan data sensor pada node <i>base</i>	51
Tabel 5. 20 Kode program <i>task</i> pencetakan data sensor pada node <i>base</i>	52
Tabel 6. 1 Hasil perbandingan sensor DHT11 dengan termometer ruangan	54
Tabel 6. 2 Output kondisi kipas.....	54
Tabel 6. 3 Output kondisi LED	55

Tabel 6. 4 Analisis hasil pengujian pengiriman dan penerimaan data sensor	56
Tabel 6. 5 Analisis hasil pengujian metode Preemptive Priority Based-Scheduling	57



DAFTAR PERSAMAAN

(6. 1)	54
(6. 2)	56



BAB 1 PENDAHULUAN

Pada bab ini dijelaskan tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan penelitian dari *"Implementasi Sistem Operasi Real-Time pada Arduino Nano dengan media komunikasi NRF24L01 untuk pengukuran suhu, kelembaban dan intensitas cahaya."*

1.1 Latar belakang

Perkembangan teknologi pada mikroprosesor saat ini dituntut untuk dapat dipasang lebih dari satu sensor. Dengan dipasangnya sensor lebih dari satu maka dibutuhkan juga banyak *task* sesuai dengan sensor yang ada guna untuk mengeksekusi sensor-sensor yang terpasang dalam sistem. Dalam proses eksekusi task-task yang ada satu secara simultan perlu diterapkannya sistem multitasking. Dengan melakukan penerapan multitasking maka program yang dapat dieksekusi dalam sistem dapat dieksekusi lebih dari satu secara bersamaan (Wahyudin, 2018). Salah satu contoh penerapan multi-sensor adalah pada sistem rumah cerdas. Sensor-sensor yang diletakkan pada rumah cerdas digunakan untuk mengontrol suhu, pencahayaan, serta banyak fungsi lainnya. Dengan berjalannya sensor-sensor itu secara simultan maka kinerja dari sistem monitoring rumah cerdas akan berjalan dengan semestinya. Jika tidak, maka sistem monitoring rumah cerdas akan bekerja tidak semestinya seperti mengalami penurunan atau kenaikan suhu yang secara signifikan dikarenakan pemrosesan data tidak tepat pada waktunya serta terlambat dalam pengambilan keputusan yang seharusnya tidak terjadi pada sistem multi-sensor.

Dalam membangun sebuah sistem *monitoring* rumah cerdas, diperlukan adanya komunikasi pada jaringan lokal. Komunikasi pada jaringan lokal ini digunakan sebagai sarana komunikasi antara sistem rumah cerdas dengan si pemilik rumah dalam menyampaikan informasi yang didapat oleh sensor. Sarana komunikasi yang sering ditemukan dalam membangun sebuah sistem monitoring rumah cerdas adalah jaringan nirkabel. Jaringan nirkabel yakni jaringan yang berfungsi untuk menghubungkan alat satu dengan alat lainnya tanpa menggunakan kabel (cisco, 1984). Modul komunikasi nirkabel yang digunakan adalah nRF24L01. Modul nRF24L01 adalah modul yang digunakan sebagai media komunikasi tanpa kabel. Modul ini menggunakan pita radio *frequency* sebagai komunikasinya (Suprayogi, 2017). Kelebihan dari penerapan modul nRF24L01 pada sistem monitoring rumah cerdas dibandingkan modul komunikasi lainnya yakni harganya yang murah, jangkauan komunikasi yang lebih baik (Shobrina, 2018), serta konsumsi daya yang rendah.

Penerapan sistem multitasking untuk menjalankan banyak task secara simultan pada sistem embedded biasanya diproses menggunakan sebuah sistem operasi. Sistem operasi yang digunakan haruslah bersifat real-time karena sebuah sistem monitoring rumah cerdas diharapkan dapat membuat keputusan dan memberikan informasi yang tepat dalam waktu yang aktual. Sistem embedded

sendiri telah mengembangkan sistem operasi yang bersifat real-time. Sistem operasi ini disebut dengan *Real Time Operating System*(RTOS). Pada RTOS, waktu bisa diprediksi dan tetap konsisten walaupun tugas yang dikerjakan bertambah (Suparlin, 2018).

Berdasarkan penjelasan di atas, dengan diterapkannya RTOS pada sistem monitoring rumah cerdas, diharapkan *task-task* yang ada dapat dieksekusi dan diprediksi dengan waktu yang aktual sehingga pengambilan keputusan dan pemberian informasi pada sistem tidak terlambat. Multi sensor yang berjalan secara simultan menerapkan RTOS dengan 2 sensor yang berbeda yaitu sensor LDR dan sensor DHT11. Sensor DHT11 akan diberi sebuah kipas dan sebuah LED sebagai indikator bahwa sensor DHT11 tersebut mendeteksi suhu yang telah ditentukan sebelumnya. Modul komunikasi nirkabel yang digunakan sebagai alat komunikasinya adalah nRF24L01. Mikrokontroler yang digunakan sebagai pemrosesnya adalah Arduino Nano. Monitoring suhu, kelembaban dan cahaya ditampilkan di serial monitor pada aplikasi Arduino IDE.

1.2 Rumusan masalah

Berdasarkan uraian pada latar belakang, rumusan masalah yang didapat adalah sebagai berikut :

1. Bagaimana implementasi sensor node menggunakan sensor DHT11, sensor LDR, dan modul *wireless* nRF24L01 pada Arduino Nano?
2. Bagaimana cara mengimplementasikan RTOS ke sebuah sistem node agar berjalan secara simultan?
3. Bagaimana metode *Preemptive Priority Based-Scheduling* diterapkan pada sistem node berbasis RTOS?

1.3 Tujuan

Berdasarkan rumusan masalah yang tertera di atas, maka tujuan dari penelitian ini dapat dijelaskan sebagai berikut :

1. Untuk mendapatkan data sensor yang sesuai menggunakan sensor DHT11 dan sensor LDR serta terjadinya komunikasi yang baik antara node base dengan node client menggunakan modul *wireless* nRF24L01.
2. Untuk membuat kode program RTOS supaya dapat diimplementasikan ke dalam sistem node sehingga sistem berjalan secara simultan.
3. Untuk mengetahui penerapan metode *Preemptive Priority Based-Scheduling* pada sistem node berbasis RTOS.

1.4 Manfaat

Diharapkan dari penelitian ini didapatkan manfaat dari beberapa pihak yang terkait dengan penelitian ini antara lain :

1. Bagi penulis :

- a. Manfaat jangka pendek yaitu sebagai dasar dalam penyusunan skripsi sehingga data yang diperoleh akan akurat.
- b. Manfaat jangka panjang yaitu menambah wawasan dan ilmu pengetahuan akan hasil penelitian yang telah dilakukan terkait topik yang diambil.
- c. Membuat sebuah sistem yang dapat melakukan multitasking dan setiap operasinya dilakukan secara *real-time*.

2. Bagi mahasiswa lain :

Untuk membantu mahasiswa lainnya dalam mengembangkan topik penelitian terkait sebagai dasar untuk penelitian-penelitian selanjutnya.

1.5 Batasan masalah

Batasan masalah pada penelitian ini agar tidak menyimpang dari perumusan masalah yang sudah ditentukan adalah sebagai berikut :

1. Terdapat 1 node yang berfungsi sebagai node *base* dan dihubungkan dengan 1 node yang berfungsi sebagai node *client*.
2. Pada node *base* menggunakan mikrokontroler Arduino Nano dan modul *wireless* nRF24L01.
3. Pada node *client* menggunakan mikrokontroler Arduino Nano, sensor DHT11, sensor LDR, dan modul *wireless* nRF24L01.
4. Implementasi *Preemptive Priority-Based Scheduling* pada setiap task baik node *client* maupun node *base* menggunakan *Real-Time Operating System*.
5. Sensor yang digunakan adalah sensor DHT11 dan sensor LDR.
6. Modul komunikasi *wireless* yang digunakan adalah nRF24L01.
7. Data yang dikirim oleh node *client* berupa nilai suhu dan kelembaban menggunakan sensor DHT11, serta nilai intensitas cahaya menggunakan sensor LDR.

1.6 Sistematika pembahasan

Penulisan skematik pembahasan dan penyusunan proposal skripsi dapat diuraikan sebagai berikut :

BAB 1 PENDAHULUAN

Pada bab ini dideskripsikan mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan penelitian dari "Implementasi Sistem Operasi Real-Time pada Arduino Nano dengan media Komunikasi Modul NRF24L01 Untuk Pengukuran Suhu dan Kelembaban."

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan tentang LANDASAN teori yang terkait dengan penelitian yang dilakukan penulis, serta menjelaskan penelitian-penelitian serupa yang sebelumnya pernah dilakukan.

BAB 3 METODOLOGI

Bab ini berisi tentang analisis kebutuhan sistem, gambaran umum perancangan sistem, implementasi sistem, pengujian sistem, serta analisis dan pembahasan sistem.

BAB 4 REKAYASA KEBUTUHAN

Bab ini dijelaskan tentang gambaran umum sistem, kebutuhan sistem yang meliputi kebutuhan fungsional, kebutuhan non fungsional, kebutuhan perangkat keras, dan kebutuhan perangkat lunak, serta batasan desain sistem.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini berisi perancangan sistem yang meliputi pendefinisian dan perancangan sistem, perancangan hardware yang berisi tentang perancangan sistem node base dan node client, perancangan prioritas pada task dan implementasi metode *preemptive priority based-scheduling*.

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini berisikan hasil pengujian dan analisis sensor DHT11, pengiriman data sensor menggunakan media komunikasi *wireless nRF24L01*, dan implementasi metode *Preemptive Priority-Based Scheduling*.

BAB 7 PENUTUP

Bab ini berisi tentang kesimpulan dan saran yang diperoleh dari rumusan masalah penelitian dengan melakukan analisis dan pengujian.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi tentang tinjauan pustaka yang digunakan sebagai acuan penelitian yang dilakukan dan pembandingan penelitian sebelumnya dengan penelitian yang dilakukan. Dasar teori berisi tentang penjelasan teori yang digunakan untuk membangun penelitian ini.

2.1 Tinjauan Pustaka

Pada penelitian ini, tinjauan pustaka sebagai referensi berasal dari jurnal yang terkait dengan topik akuisisi data sensor menggunakan RTOS (*Real-Time Operating System*), yaitu menggunakan referensi penelitian yang berjudul “Rancang Bangun Sistem Multi-Sensor Untuk Pengukuran Jarak Secara Simultan” oleh (Wahyudin, 2018). Pada penelitian tersebut memfokuskan pada penerapan RTOS pada sistem menggunakan 8 buah sensor HC-SR04 yang bekerja secara simultan untuk mengukur jarak (Wahyudin, 2018). Pada penelitian ini diusulkan untuk menggunakan 2 buah sensor dan ditambahkan alat komunikasi wireless sebagai media komunikasi node client dengan node base.

Penelitian yang kedua berjudul “Implementasi Real Time pada Pergerakan Robot Quadruped menggunakan Multisensor dan RTOS” oleh (Saputro, 2018). Pada penelitian tersebut diterapkan RTOS untuk menggerakkan robot menggunakan sensor Ultrasonik sebagai pengukur jarak (Saputro, 2018). Pada penelitian digunakan 2 buah sensor yang berbeda dan penambahan alat komunikasi wireless untuk mengetahui fungsi daripada RTOS sendiri dalam melakukan penjadwalan *task* secara *preemptive*.

Tabel 2. 1 Tabel Perbandingan Penelitian

No	Judul	Penjelasan	Tahun	Perbedaan dengan penelitian sebelumnya
1.	Rancang Bangun Sistem Multi-Sensor Untuk Pengukuran Jarak Secara Simultan	Penelitian ini menerapkan RTOS pada Arduino Mega 2560 untuk mengontrol 8 buah sensor HC-SR04 sebagai sensor jarak. Dan juga ditambahkan LED berwarna merah sebagai indikator jarak minimal yang terdeteksi.	2018	Penelitian ini menggunakan 2 sensor dengan waktu eksekusi yang berbeda dan ditambahkan media komunikasi wireless antara node client dengan node base

2.	Implementasi Real Time pada Pergerakan Robot Quadruped menggunakan Multisensor dan RTOS	Penelitian ini menerapkan RTOS pada Arduino Mega untuk mengontrol 8 buah sensor Ultrasonik sebagai sensor jarak dan menggerakkan kaki robot. Data Sensor Ultrasonik digunakan sebagai parameter untuk menghindari tabrakan dengan objek disekelilingnya	2018	Penelitian ini menggunakan 2 sensor dengan waktu eksekusi yang berbeda dan ditambahkan media komunikasi wireless antara node client dengan node base
----	---	---	------	--

2.2 Dasar Teori

Dalam sub bab ini akan dijelaskan referensi dasar teori sebagai pengetahuan tentang komponen dan teknologi yang digunakan meliputi sensor pengukuran intensitas cahaya, sensor pengukuran suhu dan kelembaban, relay, mikrokontroler Arduino Nano, komunikasi modul wireless nRF24L01, RTOS, FreeRTOS, dan metode *Preemptive Priority-Based Scheduling*.

2.2.1 Sensor Pengukuran Intensitas Cahaya

LDR(*Light Dependent Resistor*) adalah sebuah resistor nilai resistansinya berubah sesuai dengan intensitas cahaya yang diterimanya.



Gambar 2. 1 Sensor LDR

(sumber : www.components101.com)

Berdasarkan gambar 2.1, dapat dijelaskan bahwa sensor LDR sendiri terdiri dari 2 buah elektroda pada permukaan sebuah cakram semikonduktor seperti kadmium fufida. Ketika intensitas cahaya yang ditangkap oleh LDR sedikit, jumlah elektron yang dihasilkan oleh bahan dari cakram semikonduktor tersebut relatif kecil, sehingga muatan elektrik yang dapat diangkut oleh elektron berjumlah sedikit. Dapat diartikan bahwa saat intensitas cahaya yang diterima oleh sensor LDR sedikit maka sensor LDR akan memiliki jumlah resistansi yang besar. Begitupun sebaliknya, saat intensitas cahaya yang diterima oleh LDR banyak maka sensor LDR akan memiliki nilai resistansi yang kecil karena dengan banyaknya elektron bebas, maka muatan listrik dapat dialirkan lebih mudah sehingga sensor

LDR akan memiliki nilai resistansi yang kecil dan dapat menjadi konduktor yang baik.

Tabel 2. 3 Spesifikasi Sensor LDR

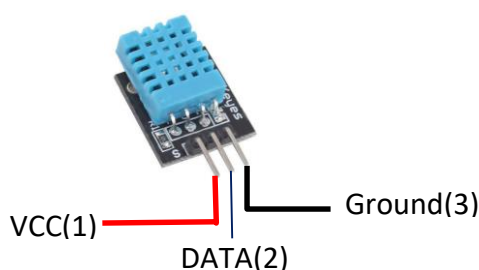
Electrical Characteristics					
Parameter	Conditions	Min	Typ	Max	Unit
Cell Resistance	1000 LUX	-	400	-	Ohm
	10 LUX	-	9	-	K Ohm
Dark Resistance	-	-	1	-	M Ohm
Dark Capacitance	-	-	3,5	-	pF
Rise Time	1000 LUX	-	2,8	-	Ms
	10 LUX	-	18	-	Ms
Fall Time	1000 LUX	-	48	-	Ms
Voltage AC/DC Peak			-	320	V max
Current			-	75	mA max
Power Dissipation				100	mW max
Operating Temperature		-60	-	75	Deg/ C

(sumber : www.suron.com)

Berdasarkan tabel 2.1, dapat dijelaskan bahwa sensor LDR dapat mengukur intensitas cahaya yang diterima dari 10 LUX sampai dengan 1000 LUX. Batasan tegangan AC/DC terbesar adalah 320 V. Arus terbesar yang dapat dialirkan adalah 75 mA. Sensor LDR dapat dioperasikan pada suhu -60 derajat celcius sampai dengan 75 derajat celcius.

2.2.2 Sensor Pengukur Suhu dan Kelembaban

Sensor DHT11 merupakan sensor digital yang dilengkapi dengan fitur pengukur suhu dan kelembaban di sekitarnya.



Gambar 2. 2 Sensor DHT11

(sumber : www.components101.com)

Berdasarkan gambar 2.2, dapat dijelaskan bahwa Sensor DHT11 memiliki 3 pin yaitu VCC ditunjukkan pada pin 1, Data ditunjukkan pada pin 2, dan Ground ditunjukkan pada pin ke 3. Sensor LDR membutuhkan daya sebesar 3,5V-5,5V. Sensor LDR dipastikan memiliki tingkat kehandalan yang tinggi serta stabilitas jangka panjang yang baik. Koefisien kalibrasi disimpan sebagai program didalam OTP memory, dimana koefisien yang telah dikalibrasi akan digunakan oleh internal sensor dalam proses mendeteksi suhu dan kelembaban. Dengan ukuran DHT11 yang relatif kecil, memakai daya yang rendah, dan memiliki transmisi sinyal sampai 20 meter, membuatnya menjadi pilihan yang terbaik untuk digunakan dalam berbagai macam implementasi pada sistem sebagai sensor pengukuran suhu dan kelembaban.

Tabel 2. 4 Spesifikasi Sensor DHT11

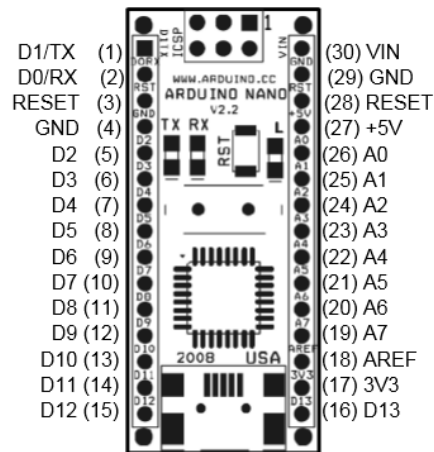
DHT11 Specifications				
Parameter	Conditions	Min	Max	Unit
Operating Voltage		3,5	5,5	Volt
Operating Current	Measuring	0,3	-	mA
	Standby	60		uA
Temperature Range		0	50	Celcius
Humidity Range		20	90	Percentage
Accuracy	Measuring	± 1	-	Celcius
		± 1		Percentage

(Sumber : www.101components.com)

Berdasarkan tabel 2.2, dapat dijelaskan bahwa sensor DHT11 menggunakan daya sebesar 3,5V sampai 5,5V. Sensor ini bekerja pada 0,3mA ketika mengukur dan 60uA ketika standby. Output data dari sensor ini berupa serial. Suhu yang dapat diukur adalah 0 derajat Celcius sampai 50 derajat Celcius. Tingkat kelembaban yang dapat diukur adalah 20% sampai 90%. Untuk penghitungan suhu dan kelembaban keduanya adalah 16 bit. Tingkat selisih akurasi dari pengukuran suhu dari suhu yang sebenarnya adalah kurang lebih 1 derajat Celcius sedangkan pada kelembaban adalah kurang lebih 1%.

2.2.3 Mikrokontroler Arduino Nano

Arduino Nano adalah development board berupa mikrokontroler yang diimplementasikan chip Atmega328 dengan bentuk yang mini. Fungsinya tidak berbeda jauh dengan Arduino Uno. Perbedaan utama antara Arduino Nano dengan Arduino Uno adalah tidak adanya jack power DC dan konektor Mini-B USB.



Gambar 2. 3 Arduino Nano

(Sumber :www.components101.com)

Berdasarkan gambar 2.3, dapat dijelaskan bahwa Arduino Nano disebut sebagai papan pengembangan (development board) karena board ini memiliki fungsi sebagai mikrokontroler untuk mengontrol sistem implementatif. Akan lebih mudah menggunakan development board dalam merangkai rangkaian elektronika daripada mulai merakit menggunakan mikrokontroler ATmega328 dari awal di breadboard.

Berikut penjelasan pin-pin yang terdapat gambar 2.3 :

a.) Power Supply

Development Board Arduino Nano mendapatkan daya dari koneksi kabel mini-B USB atau melalui daya eksternal menggunakan baterai. Penyedia daya eksternal bisa dihubungkan dengan pin 30 atau Vin (Unregulated 6V – 20V), atau pin 27 (regulated 5V).

Beberapa pin power yang terdapat pada Arduino Nano :

- Pin GND adalah pin yang berfungsi sebagai ground dan bernilai nol.
- Pin Vin adalah pin yang dihubungkan dengan penyedia power eksternal secara langsung untuk board arduino.
- Pin 5V adalah pin output pemberi aliran tegangan sebesar 5V yang telah diproses melalui regulator.
- Pin 3V3 adalah pin output pemberi aliran tegangan sebesar 3.3V yang telah melalui regulator.
- Pin REF adalah pin yang berfungsi sebagai penyedia referensi tegangan mikrokontroler. Biasanya digunakan pada board shield untuk memperoleh tegangan yang sesuai antara 5V atau 3.3V.

b.)Memori

Chip ATmega328 hanya memiliki memori sebesar 32 KB dan hanya sebesar 0,5 KB dari jumlah memori yang ada untuk digunakan sebagai bootloader. SRAM menggunakan memori sebesar 2 KB sedangkan 1KB untuk EEPROM. Baca tulis pada saat melakukan pemrograman dapat menggunakan EEPROM *library*.

c.) Input dan Output(I/O)

Arduino Nano memiliki input dan output sebanyak 14 buah digital pin. Cara menggunakan input ataupun output, dapat digunakan fungsi *digitalWrite()*, *digitalRead()*, dan *pinMode()*.

Berikut adalah fungsi khusus dari beberapa pin :

- Pin **Serial** : Terdiri dari pin 2(RX) dan pin 1(TX). Pin RX berfungsi untuk menerima data serial sedangkan pin TX berfungsi untuk mengirim data serial.
- Pin **External Interrupts** : Terdiri dari pin 2 dan pin 3. Kedua pin ini digunakan untuk mengaktifkan interrupt dengan fungsi *attachInterrupt()*.
- Pin **PWM** : Terdiri dari pin 3, 5, 6, 9, 10, dan 11 yang menyediakan output PWM dengan fungsi *analogWrite()*.
- Pin **SPI** : Terdiri dari pin 10(SS), 11(MOSI), 12(MISO), dan 13(SCK) yang mendukung komunikasi SPI dengan menggunakan *SPI library*.
- Pin **LED** : Terdapat pada pin 13 yang terhubung dengan *built-in LED* yang dikendalikan oleh digital pin no 13.

Arduino Nano memiliki 8 buah pin input untuk analog yang terdiri atas pin A0 hingga A7. Arduino Nano juga memiliki beberapa pin lain antara lain pin I2C yang terdiri dari pin A4(SDA) dan A5(SCL) untuk mendukung komunikasi I2C(TWI) dengan menggunakan *Wire library*, pin AREF yang digunakan sebagai referensi tegangan untuk input analog, dan pin Reset yang digunakan untuk melakukan reset pada mikrokontroler.

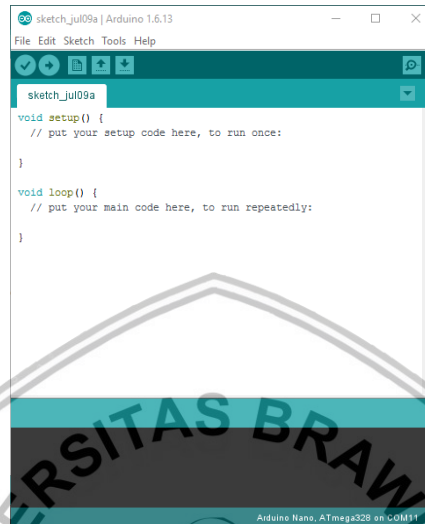
d.)Komunikasi

Arduino Nano memiliki beberapa macam alternatif untuk berkomunikasi dengan komputer, berkomunikasi dengan arduino lainnya, atau dengan mikrokontroler lainnya. Chip ATmega328 menyediakan komunikasi serial UART TTL(5V) yang tersedia pada pin 2(RX) dan pin 1(TX). Pada bagian komunikasi dengan komputer, terdapat Software Arduino IDE yang dapat menampilkan data serial dengan menggunakan Serial Monitor.

2.2.4 Arduino IDE

IDE(Integrated Development Environment) merupakan perangkat lunak yang biasa digunakan dalam *development* pada Arduino maupun mikrokontroler yang

biasa digunakan Arduino yaitu ATmega328 dan sejenisnya. Perangkat lunak ini memiliki bahasa pemrograman sendiri yang mana bahasa tersebut pengembangan dari bahasa pemrograman C yang disebut Sketch. Arduino IDE dikembangkan dari bahasa pemrograman JAVA serta dilengkapi dengan beberapa library pendukung dalam pengembangan Arduino.



Gambar 2. 4 Tampilan software Arduino IDE

(Sumber : www.Arduino.cc)

Berdasarkan pada gambar 2.4, Arduino IDE memiliki beberapa bagian, antara lain:

- **Verify**

Fungsi ini digunakan untuk mengecek apakah ada kesalahan dalam penulisan syntax atau tidak.

- **Upload**

Fungsi ini digunakan untuk mengirim kode program ke dalam board Arduino.

- **New**

Fungsi ini digunakan untuk membuat Sketch baru

- **Open**

Fungsi ini digunakan untuk membuka kode program Arduino(Sketch) yang sudah ada.

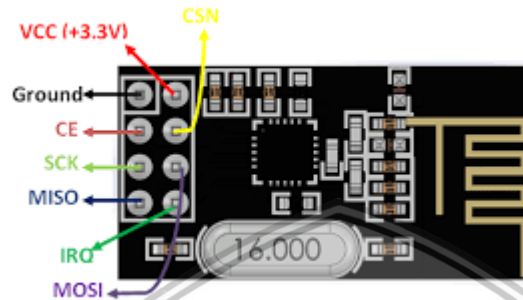
- **Save**

Fungsi ini digunakan untuk menyimpan Sketch.

- **Serial Monitor**

Fungsi ini digunakan untuk menampilkan program atau melakukan debugging melalui Arduino IDE. Serial monitor ini digunakan untuk melihat alur program maupun sebuah nilai yang ingin ditampilkan pada program untuk mengecek kinerja dari program tersebut.

2.2.5 Modul *Wireless* NRF24L01



Gambar 2. 5 Modul *wireless* nRF24L01

(sumber : www.arduino.cc)

Berdasarkan gambar 2.5, dapat kita lihat bahwa modul wireless yang menggunakan mikrokontroler NRF24L01 memiliki kaki pin sebanyak 8 buah antara lain Ground, CE, SCK, MISO, IRQ, MOSI, CSN, dan VCC. Gelombang frekuensi yang digunakan NRF24L01 sebesar 2,4GHz ISM(Industrial, Scientific and medical) dengan kecepatan mencapai 2Mbps dan memiliki pilihan data rate 250Kbps, 1Mbps dan 2Mbps. Modul ini bisa menjangkau sejauh 1000M karena modul ini dilengkapi dengan *Power Amplifier* dan *LOW Noise Amplifier* sehingga dapat melakukan transfer data yang cukup jauh dan stabil. NRF24L01 bekerja dengan tegangan sebesar 3,0V-3,6V.

Interface SPI(*Serial Peripheral Interface*) adalah *interface* yang digunakan oleh modul *wireless* nRF24L01 sebagai komunikasi dimana SPI adalah salah satu jenis protokol komunikasi yang bersifat *full duplex*, yaitu adanya device yang bertindak sebagai *Master* dan device lainnya sebagai *Slave*.

2.2.6 Lampu LED



Gambar 2. 6 Lampu LED

(sumber : www.electronics-tutorial.ws)

Berdasarkan gambar 2.6, dapat dijelaskan bahwa LED atau kepanjangannya yaitu light Emitting Diode adalah sebuah komponen elektronika yang ketika diberikan tegangan maju maka dapat memancarkan cahaya monokromatik. LED terbuat dari bahan-bahan semikonduktor. Jenis bahan semikonduktor yang digunakan mempengaruhi warna-warna yang dipancarkan oleh LED. Sinar inframerah yang tidak terlihat oleh mata dapat dipancarkan oleh LED dimana sering terdapat pada Remote Control perangkat elektronik. LED tidak memerlukan pembakaran filamen seperti lampu pijar sehingga LED tidak menghasilkan panas ketika dalam keadaan menyala.

2.2.7 Kipas Motor DC



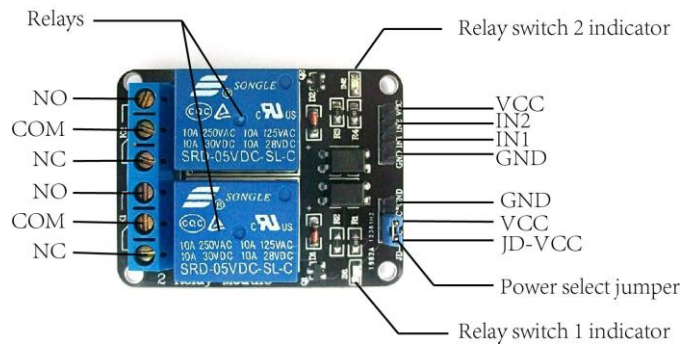
Gambar 2. 7 Kipas Motor DC

(sumber : www.sparkfun.com)

Berdasarkan gambar 2.7, Kipas Motor DC adalah suatu perangkat dimana energi listrik diubah menjadi energi kinetik atau gerakan. Kipas Motor DC ini sering disebut sebagai Kipas Motor arus searah. Kipas Motor DC ini memiliki dua terminal dan memerlukan tegangan arus searah atau DC (Direct Current) untuk dapat menggerakkan baling-balingnya.

Kipas Motor Listrik DC ini dapat berputar searah jarum jam maupun berlawanan arah jarum jam tergantung polaritas listrik yang diberikan pada kipas motor DC tersebut. Perputaran yang dihasilkan permenit oleh pada Kipas Motor Listrik DC disebut juga dengan RPM (Revolutions per minute).

2.2.8 Relay



Gambar 2. 8 Relay 2 Channel

(sumber : www.sparkfun.com)

Berdasarkan gambar 2.8, dapat dijelaskan bahwa Relay merupakan komponen listrik berupa saklar yang digunakan untuk mengatur aliran listrik serta mengubah dan menghantarkan arus listrik yang kecil menjadi arus listrik yang lebih tinggi. Contohnya yaitu Relay yang menggunakan Elektromagnet 5V dan 50mA dapat menghantarkan listrik 220V dengan arus 2A.

2.2.9 Sistem Operasi Real-Time

Real Time Operating System (RTOS) adalah sistem operasi yang mengeksekusi *task-task* dalam sebuah pola yang teratur dan telah ditentukan. Sistem kerja RTOS berbeda dengan sistem kerja OS pada umumnya. Sistem kerja OS pada umumnya akan bekerja sesaat setelah aliran listrik masuk kepada CPU (*Central Processing Unit*), kemudian Sistem Operasi akan menjalankan program-program yang ada. Sedangkan pada RTOS, sistem operasi ini dikomodasi oleh sebuah program otomatis. Program ini adalah sebuah kernel. Kernel akan menyala terlebih dahulu kemudian baru menyalakan RTOS ketika sistem dinyalakan. Tugas-tugas yang terdapat pada RTOS antara lain mengatur sumber daya yang meliputi prosesor dan memori/register, serta hak akses untuk mengakses prosesor dan memori. Selain itu, RTOS juga bertugas untuk melakukan komunikasi dan sinkronisasi pada sistem.

Real Time Operating System (RTOS) mempunyai beberapa konsep yang diterapkan dalam sistem operasi, 2 di antaranya adalah *Multitasking* dan *Scheduling*. *Multitasking* dan *Scheduling* dalam RTOS dapat dijelaskan sebagai berikut :

- **Multitasking**

Sistem operasi mempunyai beberapa komponen untuk mengatur operasi *task* secara multitasking. *Multitasking* sendiri adalah kemampuan sistem operasi untuk mengoperasikan lebih dari satu *task* dalam satu waktu. Setiap *task* yang terdapat dalam kode program terbentuk dari setiap proses yang harus dijalankan oleh sebuah *embedded system*. Sistem operasi mampu untuk

menelusuri *task-task* yang dijalankan dari satu *task* ke *task* lainnya tanpa kehilangan informasi dari *task* yang dijalankan.

- **Scheduling**

Scheduling adalah konsep pembagian waktu eksekusi *task-task* yang ada. Scheduling sendiri merupakan bagian terpenting dalam melakukan sebuah multiprogramming di sistem operasi. Proses dalam melakukan scheduling yakni menghapus *task* yang berjalan dan memilih *task* yang lain sesuai dengan pembagian jadwal *task* yang telah ditentukan. Untuk menentukan waktu eksekusi *task* diperlukan suatu algoritma. Salah satu algoritma scheduling dalam sistem operasi real-time yang sering digunakan yaitu algoritma Preemptive Priority-Based Scheduling.

2.2.10 Free Sistem Operasi Real-Time

FreeRTOS (Free Real-time operating systems) adalah sebuah sistem operasi *real-time* yang bersifat open source untuk sistem embedded. FreeRTOS ini banyak digunakan oleh mikrokontroler untuk kebutuhan sistem operasinya. Beberapa varian mikrokontroler yang menggunakan FreeRTOS antara lain: ARM architecture, Atmel AVR, AVR32, PIC Microcontrollers, MicroBlaze, Arduino, dan masih banyak lagi.

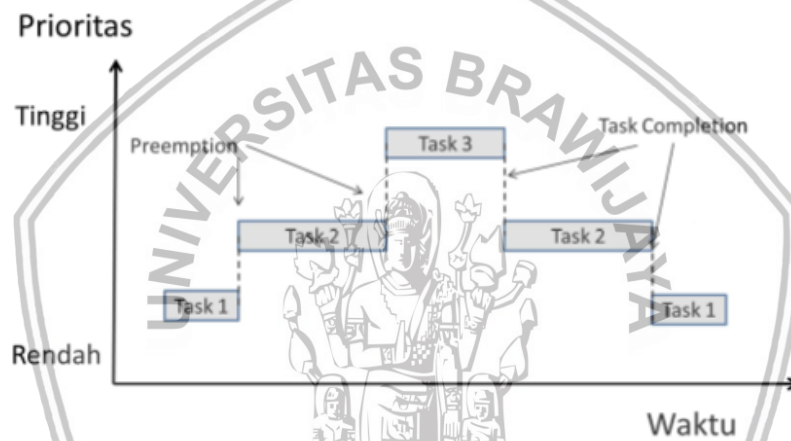
Yang dimaksud dengan RTOS (Real Time Operating System) adalah sistem operasi yang berfungsi untuk menjalankan *task-task* yang ada dan bersifat *real-time*. Yang dimaksud dengan *real-time* adalah waktu yang sebenarnya di mana *task* atau kejadian tertentu terjadi. Salah satu keuntungan dalam menggunakan RTOS adalah kemampuannya untuk mengeksekusi *task* yang ada sesuai dengan jadwal dan prioritas yang telah ditentukan secara konsisten.

Desain yang sering digunakan untuk RTOS biasanya mengandung :

- 1.) Event driven, pemrograman di dalam FreeRTOS dan RTOS pada umumnya berlandaskan pada event-driven, yaitu program berjalan berdasarkan event-event yang ada dan telah didefinisikan sebelumnya. Event-driven erat kaitannya dengan sensor dan interaksi user dalam pengerjaannya.
- 2.) Time-sharing, yang mempunyai tugas untuk mengontrol *clock interrupt* dan *on event* atau biasa dikenal dengan algoritma *round robbin* (sebuah algoritma penjadwalan *task*). Algoritma penjadwalan ini juga sering digunakan pada FreeRTOS. *Time-sharing* adalah sebuah sistem operasi yang dibuat dalam skala kecil dan sederhana. Kernel FreeRTOS hanya terdiri dari tiga atau empat file C (biasanya kernel FreeRTOS dibangun dari bahasa C). Penggunaan bahasa C digunakan agar memudahkan untuk code readable, memudahkan koneksi ke port, dan mudah dalam melakukan perawatan.

2.2.11 Metode Preemptive Priority-Based Scheduling

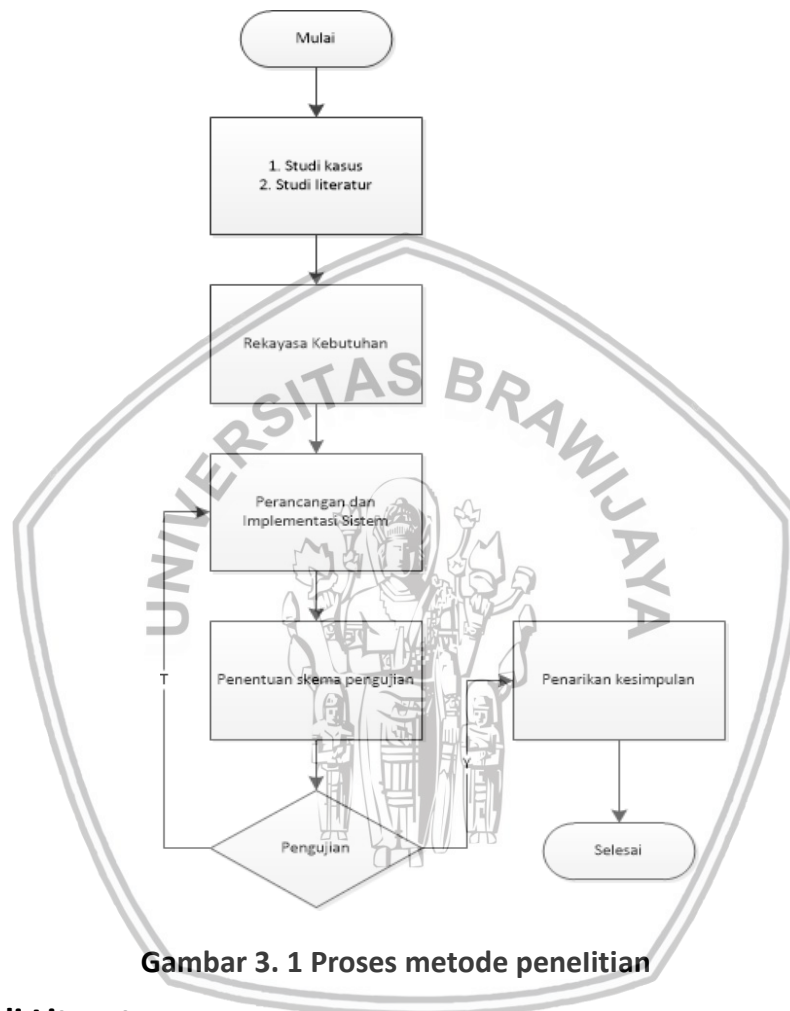
Salah satu algoritma *scheduling* di dalam RTOS yaitu algoritma *Preemptive Priority-Based Scheduling*. Algoritma *Preemptive Priority-Based Scheduling* mengatur setiap waktu eksekusi *task* dimulai dari prioritas yang paling tinggi sampai yang terendah. Pada gambar 2.9, dapat dijelaskan algoritma *Preemptive Priority-Based Scheduling* bekerja sesuai dengan prioritas *task* yang tertinggi hingga yang terendah. *Task* yang memiliki prioritas paling tinggi akan dijalankan terlebih dahulu, sedangkan *task* dengan prioritas yang lebih rendah akan menunggu *task* yang memiliki prioritas paling tinggi selesai dijalankan. Kemampuan dalam pemberian prioritas *task* secara dinamis ini menambah fleksibilitas sebuah *embedded system* terhadap *event-event* tertentu yang menciptakan suatu kondisi dalam *embedded system* bersifat *real-time* dan *responsive*.



Gambar 2. 9 Algoritma *Preemptive Priority-Based Scheduling*

BAB 3 METODOLOGI

Bab ini menjelaskan tentang alur metode penelitian, perancangan penelitian, pengujian sistem, analisis dan pembahasan, serta proses pembuatan laporan penelitian.



Gambar 3. 1 Proses metode penelitian

3.1 Studi Literatur

Studi literatur pada sub bab ini bertujuan untuk mengkaji hal-hal terkait teori-teori yang mendukung dalam perencanaan dan implementasi alat. Teori-teori yang telah dikaji adalah sebagai berikut :

1. Metode *Preemptive Priority-Based Scheduling*

Pada bagian ini dilakukan kajian terkait dengan penelitian terdahulu yang telah menerapkan metode *Preemptive Priority-Based Scheduling* sebagai metode penjadwalan *sensing* data sensor cahaya, sensor suhu dan kelembaban serta pengiriman data pada node *client*, juga penjadwalan penerimaan data dan menampilkan data yang telah dikirim melalui serial monitor.

2. Modul Komunikasi *Wireless*

Bagian ini dilakukan penelusuran literatur yang berhubungan dengan cara penggunaan modul komunikasi *wireless* yang digunakan agar dapat diimplementasikan pada mikrokontroler yang digunakan.

3. Mikrokontroler

Bagian ini dilakukan penelusuran literatur yang berhubungan dengan cara penggunaan mikrokontroler yang digunakan agar dapat diimplementasikan pada sistem.

4. Sensor Cahaya

Bagian ini dilakukan penelusuran literatur yang berhubungan dengan cara penggunaan sensor cahaya yang digunakan agar dapat diimplementasikan pada mikrokontroler yang digunakan.

5. Sensor Suhu dan Kelembaban

Bagian ini dilakukan penelusuran literatur yang berhubungan dengan cara penggunaan sensor suhu dan kelembaban yang digunakan agar dapat diimplementasikan pada mikrokontroler yang digunakan.

6. Relay

Bagian ini dilakukan penelusuran literatur yang berhubungan dengan cara penggunaan relay yang digunakan agar dapat diimplementasikan pada mikrokontroler yang digunakan.

3.2 Analisis Kebutuhan

Analisa kebutuhan memiliki tujuan untuk menganalisis kebutuhan dalam merancang sebuah sistem, menguji sistem dan menganalisis hasil dari sistem yang dirancang. Analisis ini diawali dengan kebutuhan perangkat keras dari sistem yang dirancang dan kebutuhan perangkat lunak yang akan diimplementasikan pada sistem. Manfaat dari membuat analisis kebutuhan yaitu memudahkan dalam membangun sebuah sistem.

3.2.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras bertujuan untuk membantu penelitian ini dalam mendefinisikan kebutuhan-kebutuhan yang digunakan dalam sistem. Alat dan bahan yang digunakan dalam implementasi sistem yang dirancang antara lain :

1. Sensor DHT11
2. Sensor LDR
3. Mikrokontroler Arduino Nano
4. Modul komunikasi *wireless* nRF24L01
5. LED

6. Kipas motor DC
7. laptop/ Komputer PC
8. Arduino IDE
9. Relay
10. Jumper
11. Resistor

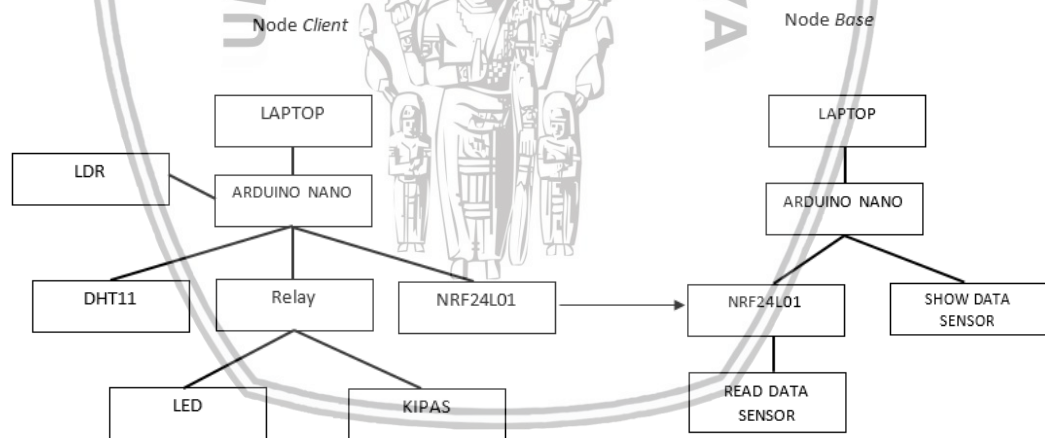
3.2.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak dalam implementasi pada sistem yang dirancang, antara lain :

- Microsoft Windows 10 64-bit sebagai sistem operasi
- Arduino IDE

3.3 Gambaran Umum Perancangan

Gambaran Umum Perancangan sistem ini dilakukan bertujuan supaya perancangan sistem penelitian ini lebih terstruktur. Perancangan implementasi sistem pada penelitian ini dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Perancangan Sistem

Berdasarkan Gambar 3.2 perancangan sistem penelitian dapat dijelaskan sebagai berikut :

1. Metode *Preemptive Priority-Based Scheduling* yang telah ditetapkan pada node client yaitu pengambilan data sensor LDR dan data sensor DHT11, setelah itu pengiriman data sensor kepada node server. Pengambilan data dan pengiriman data sensor ditampilkan melalui *serial monitor* node client. Setelah itu dilakukan pengontrolan pada lampu LED dan kipas motor DC berdasarkan data suhu yang diambil.

2. Metode *Preemptive Priority-Based Scheduling* yang telah ditetapkan pada node server yaitu penerimaan data sensor yang telah dikirim oleh node client, lalu menampilkannya pada *serial monitor* node *base*.
3. Pengaturan eksekusi waktu yang berbeda pada masing-masing task dalam node *base* dan node *client*.

3.4 Implementasi

Implementasi pada penelitian ini meliputi proses perancangan sistem penelitian sampai dengan hasil akhir penelitian. Adapun tahapan dari implementasi sistem pada penelitian ini, yaitu :

1. Implementasi perangkat keras meliputi Arduino Nano, sensor LDR, sensor DHT11, modul komunikasi *wireless* nRF24L01, lampu LED, dan kipas motor DC yang dapat dikontrol sekaligus dimonitor oleh komputer.
2. Implementasi metode *Preemptive Priority-Based Scheduling* sebagai metode penjadwalan *task* pada masing-masing node *base* dan node *client*.

Dari proses implementasi pada penelitian ini diharapkan dapat menghasilkan sistem sebagai berikut :

1. Arduino Nano yang dapat mengontrol sensor LDR, sensor DHT11, modul komunikasi *wireless* nRF24L01, LED dan kipas motor DC yang dapat dimonitor oleh komputer.
2. Pengoperasian *task* berdasarkan prioritas yang lebih tinggi.

3.5 Pengujian Sistem

Pada sub bab ini menjelaskan tentang pengujian sistem yang dilakukan sesuai dengan parameter uji coba penelitian. Beberapa rancangan skenario pengujian sistem pada penelitian ini, antara lain :

1. Pengujian pada akuisisi data sensor DHT11.
2. Pengujian pengiriman data menggunakan modul komunikasi nRF24L01 pada sisi node *Client*.
3. Pengujian penerimaan data menggunakan modul komunikasi nRF24L01 pada sisi node *Base*.
4. Pengujian eksekusi waktu penjadwalan *task* pada node *Client* dan node *Base* menggunakan metode *Preemptive Based-Priority Scheduling*.

3.6 Analisis dan Pembahasan

Pada sub bab ini dilakukan analisis serta pembahasan tentang proses penelitian yang dilakukan dan diuji. Hasil penelitian ini dianalisis untuk mengetahui apakah penelitian telah berjalan sesuai dengan yang diharapkan. Setelah dianalisis maka akan ditarik kesimpulan dari penelitian ini.

3.7 Kesimpulan

Dilakukan penarikan kesimpulan setelah proses perancangan sistem, implementasi sistem, pengujian sistem, analisis dan pembahasan sistem. Kesimpulan berisi gambaran dari hasil implementasi metode Preemptive Priority-Based. Kesimpulan adalah jawaban dari rumusan masalah yang telah ditulis pada bab pendahuluan. Pada bagian terakhir adalah penulisan saran yang berguna untuk memperbaiki dan mengembangkan sistem yang ada pada penelitian ini serta mengkaji ulang implementasi metode *Preemptive Priority-Based Scheduling* yang dirancang menggunakan sistem operasi *real-time*.

3.8 Pembuatan Laporan Skripsi

Pada tahap ini adalah tahap akhir dalam penyelesaian penelitian. Dokumentasi dari pelaksanaan skripsi dilakukan dalam penyusunan karya tulis penelitian. Penulisan karya tulis penelitian ini dilakukan sebagai penjelasan dari hasil penelitian agar mempermudah proses pengembangan untuk penelitian selanjutnya.



BAB 4 REKAYASA KEBUTUHAN

Bab ini menjelaskan secara rinci terkait deskripsi umum dari sistem, rekayasa antar-muka sistem, kebutuhan perangkat keras, kebutuhan perangkat lunak, kebutuhan fungsional, kebutuhan komunikasi, kebutuhan performansi, batasan desain sistem dan alur kerja sistem.

4.1 Gambaran Umum Sistem

Pada bagian ini dijelaskan tentang gambaran umum pada sistem. Gambaran umum pada sistem penelitian ini adalah menggunakan 2 buah node yang dihubungkan dengan media komunikasi wireless NRF24L01. Node Client berfungsi sebagai node yang digunakan untuk mengirimkan data sensor LDR dan DHT11. Node ini ditambahkan dengan relay untuk mengatur LED dan kipas yang digunakan sebagai output dari data sensor DHT11. Sedangkan pada node base digunakan sebagai node yang menerima data sensor LDR dan DHT11 yang dikirimkan oleh node client lalu dicetak dan ditampilkan melalui *serial monitor*.

4.2 Kebutuhan Sistem

Pada bagian ini berisi penjelasan kebutuhan fungsional, kebutuhan non fungsional, kebutuhan komunikasi, kebutuhan hardware dan kebutuhan software yang dianalisis sesuai dengan apa yang sistem butuhkan sehingga diharapkan dapat mempermudah dalam melakukan desain sistem dan implementasi sistem.

4.2.1 Kebutuhan Fungsional

Adapun kebutuhan fungsional yang harus dipenuhi dalam penelitian ini antara lain :

1. **Sistem dapat melakukan penjadwalan task menggunakan metode *Preemptive Priority-Based Scheduling* pada node client**

Metode *Preemptive Priority-Based Scheduling* adalah metode penjadwalan operasi *task* berdasarkan prioritas *task* yang tertinggi. Pada node *client*, prioritas *task* yang tertinggi adalah mendeteksi intensitas cahaya menggunakan sensor LDR. Prioritas *task* yang kedua adalah mendeteksi suhu dan kelembaban menggunakan sensor DHT11. Prioritas *task* yang terendah adalah mengirimkan data sensor menggunakan modul komunikasi *wireless* NRF24L01.

2. **Sistem dapat melakukan penjadwalan task menggunakan metode *Preemptive Priority-Based Scheduling* pada node base**

Metode *Preemptive Priority-Based Scheduling* adalah metode penjadwalan operasi *task* berdasarkan prioritas *task* yang tertinggi. Pada node *base*, prioritas *task* yang tertinggi adalah menerima data sensor yang dikirimkan oleh node *client*. Prioritas *task* yang terendah adalah menampilkan data sensor yang telah diterima menggunakan *serial monitor*.

3. Fungsi mendeteksi suhu dan kelembaban

Fungsi ini digunakan untuk mendeteksi suhu dan kelembaban yang ada disekitar lingkungan node menggunakan sensor DHT11.

4. Fungsi menangkap intensitas cahaya

Fungsi ini digunakan untuk menangkap intensitas cahaya yang ada disekitar node menggunakan sensor LDR.

5. Fungsi komunikasi mengirimkan data sensor

Fungsi ini terdapat pada node *client* yang digunakan untuk mengirimkan data sensor pada node *base*.

6. Fungsi komunikasi menerima data sensor *sensor* secara *real-time*

Fungsi ini terdapat pada node *base* yang digunakan untuk menerima data sensor yang dikirimkan oleh node *client*.

7. Fungsi menampilkan data sensor pada node *base*

Fungsi ini digunakan untuk menampilkan data sensor yang diterima melalui *serial monitor*.

4.2.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional menjelaskan mengenai apa saja yang menjadi batasan terhadap kebutuhan perancangan sistem. Adapun kebutuhan non fungsional yang harus dipenuhi dalam penelitian ini adalah :

1. Lingkungan Operasi

Berdasarkan lingkungan operasi sistem, ada beberapa persyaratan lingkungan yang mendukung berjalannya sistem sebagai berikut :

- **Jarak antara node *client* dengan node *base***

Peletakan posisi node *client* dengan node *base* tidak dapat berjauhan karena dapat mempengaruhi keberhasilan pengiriman dan penerimaan data.

- **Konsumsi daya dari masing-masing perangkat**

Setiap perangkat keras yang digunakan dalam sistem ini membutuhkan daya yang sesuai agar dapat bekerja dengan baik. Mikrokontroler Arduino Nano membutuhkan daya sebesar 5V yang didapatkan dari port usb yang terhubung dengan komputer atau laptop. Sensor DHT11 dan sensor LDR membutuhkan daya sebesar 5V yang dihubungkan dengan mikrokontroler Arduino Nano, sedangkan modul komunikasi *wireless* NRF24L01 membutuhkan daya sebesar 3,3V yang terhubung dengan mikrokontroler Arduino Nano.

4.2.3 Kebutuhan Perangkat Keras

Sistem node sensor pada node *base* memiliki 2 *task* yang berfungsi untuk menerima data sensor dan menampilkan data, sedangkan pada node *client* memiliki 3 *task* yang berfungsi sebagai pembacaan data sensor LDR, sensor DHT11, dan mengirimkan data sensor kepada node *base*. Supaya RTOS dapat diimplementasikan di dalam sistem dan sistem dapat dijalankan sesuai penjadwalan *task* yang telah ditentukan sebelumnya, berikut adalah kebutuhan perangkat keras yang digunakan di dalam sistem, antara lain :

1. Komputer/ Laptop

Komputer digunakan untuk memonitor sekaligus memberikan *input-output* yang dibutuhkan, yaitu input program untuk mikrokontroler dan *output* mikrokontroler yang ditampilkan dalam *serial monitor*. Adapun spesifikasi komputer yang digunakan adalah sebagai berikut :

- Model Perangkat : Asus X555O
- Processor : AMD Quad Core A10-9600P, 2,4GHz
- RAM : DDR4 8 GB

2. Sensor DHT11

Sensor DHT11 adalah sensor digital yang digunakan untuk mendeteksi suhu dan kelembaban sekitarnya.

3. Sensor LDR

Sensor LDR adalah sensor analog yang digunakan untuk menangkap intensitas cahaya yang ada disekitarnya.

4. Arduino Nano

Arduino Nano merupakan papan mikrokontroler yang digunakan sebagai perangkat pengolah data yang akan dikirimkan maupun diterima. Arduino Nano sendiri akan ditanami program sehingga dapat bekerja sesuai dengan topik penelitian.

5. Modul *Wireless* nRF24L01

nRF24L01 merupakan salah satu jenis modul komunikasi *wireless* yang digunakan untuk melakukan komunikasi data sensor antara node *client* dengan node *base*.

6. Resistor 10 kilo Ohm

Resistor adalah salah satu komponen elektronika yang berfungsi untuk membatasi atau menghambat aliran listrik yang mengalir. Resistor yang bernilai 10 kilo ohm ini digunakan pada rangkaian sensor LDR supaya aliran arus yang mengalir pada sensor LDR stabil.

7. Jumper

Jumper adalah kawat halus yang dibalut oleh karet silikon yang digunakan untuk menghantarkan listrik.

8. Breadboard mini

Breadboard mini adalah sebuah papan berukuran kecil yang digunakan untuk membuat rangkaian elektronik dalam rangkaian node *client*.

9. Relay

Relay adalah sebuah alat yang berfungsi sebagai saklar untuk menyalakan kipas dan led.

10. Kipas

Kipas adalah sebuah output pada node *client* dalam suatu kondisi yang telah ditentukan.

11. LED

LED adalah sebuah output pada node *client* dalam suatu kondisi yang telah ditentukan.

4.2.4 Kebutuhan Perangkat Lunak

Adapun kebutuhan perangkat lunak dalam penelitian ini, meliputi :

1. Microsoft Windows 10 64-bit

Parangkat lunak ini digunakan sebagai sistem operasi yang digunakan oleh komputer/ laptop.

2. Arduino IDE

Perangkat lunak ini digunakan untuk melakukan pengembangan pada mikrokontroler dengan cara menuliskan program, menjalankan program dan menerapkan program ke mikrokontroler. Perangkat lunak ini juga dapat menampilkan data yang telah diolah oleh mikrokontroler melalui *serial monitor*.

3. FreeRTOS

FreeRTOS adalah sebuah sistem operasi *real time* yang bersifat open source untuk sistem embedded. FreeRTOS ini banyak digunakan oleh mikrokontroler untuk kebutuhan sistem operasinya. FreeRTOS didistribusikan secara gratis lewat GPL dengan beberapa pengecualian untuk kalangan industri tertentu.

4.3 Batasan Desain Sistem

Dalam implementasi sistem, terdapat batasan-batasan dari desain sistem penelitian yang mana dimaksudkan agar penelitian ini lebih terarah dan berjalan sesuai harapan. Batasan-batas desain sistem tersebut, antara lain :

1. Sistem menggunakan mikrokontroler Arduino Nano.

2. Sistem menggunakan sensor LDR dan sensor DHT11.
3. Sistem menggunakan modul komunikasi *wireless* NRF24L01.
4. Sistem mengimplementasikan metode *Preemptive Based-Priority Scheduling* sebagai metode penjadwalan *task*.
5. Data sensor yang ditampilkan melalui serial monitor pada node *base* bersifat sementara atau *temporary*.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini menjelaskan tentang perancangan dan implementasi sistem berdasarkan hasil analisis dan kajian dasar teori.

5.1 Perancangan Sistem

Perancangan sistem ini dilakukan untuk merencanakan langkah-langkah yang harus dilakukan dalam melakukan penelitian. Langkah-langkah penelitian yang harus dilakukan antara lain perancangan hardware, software, dan perancangan metode-metode yang digunakan untuk mendukung berjalannya sistem ini.

5.1.1 Pendefinisian dan Perancangan Penelitian

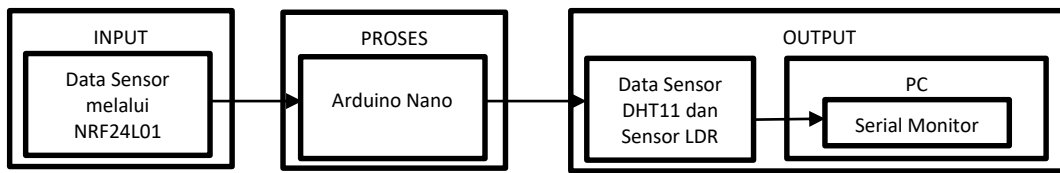
Penelitian ini dilakukan dengan mengimplementasikan metode Preemptive Priority-Based Scheduling pada Arduino Nano. Metode Preemptive Based-Priority Scheduling digunakan sebagai metode penjadwalan operasi task berdasarkan prioritas tertinggi. Metode ini digunakan untuk mengatasi permasalahan adanya penundaan task pengiriman data yang dikirimkan.

Implementasi metode Preemptive Based-Priority menggunakan sistem operasi real-time ini dilakukan dengan merangkai hardware yang dibutuhkan agar dapat bekerja secara multitasking dengan kebutuhan penelitian. Hardware yang digunakan terdiri atas mikrokontroler Arduino Nano, sensor LDR, sensor DHT11, serta modul komunikasi wireless NRF24L01. Metode Preemptive Based-Priority Scheduling diimplementasikan baik pada node client maupun node base. Penjadwalan operasi task berdasarkan prioritas yang tertinggi pada node client yaitu menangkap intensitas cahaya menggunakan sensor LDR. Operasi task selanjutnya berdasarkan prioritas yang kedua yaitu mendeteksi suhu dan kelembaban menggunakan sensor DHT11. Operasi task selanjutnya berdasarkan prioritas yang terendah yaitu mengirimkan data sensing sensor LDR dan sensor DHT11 pada node base menggunakan modul komunikasi wireless NRF24L01. Sedangkan penjadwalan operasi task berdasarkan prioritas yang tertinggi pada node base yaitu menerima data sensing sensor yang dikirimkan oleh node client menggunakan modul komunikasi wireless NRF24L01. Operasi task berikutnya berdasarkan prioritas terendah adalah menampilkan data sensing sensor yang diterima melalui serial monitor.

5.1.2 Perancangan *Hardware*

Perancangan hardware terdiri atas 2 node yaitu node base dan node client. Pada node base terdapat 2 komponen hardware yang digunakan, yaitu : Arduino Nano dan modul komunikasi wireless NRF24L01. Sedangkan pada node client menggunakan 4 komponen utama, yaitu : Arduino Nano, sensor DHT11, sensor LDR dan modul wireless NRF24L01 sebagai komunikasi. Pada node base dan node client diimplementasikan metode Preemptive Based-Priority Scheduling.

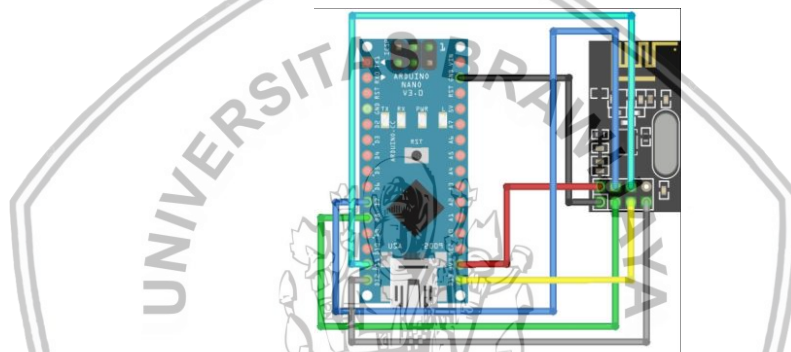
5.1.2.1 Perancangan Sistem Node Base



Gambar 5. 1 Diagram blok node base

Pada gambar 5.1, dapat dijelaskan bahwa diagram blok pada node base :

1. Hasil dari data sensor akan diterima melalui media komunikasi nRF24L01.
2. Proses data dilakukan oleh Arduino Nano.
3. Output data sensor ditampilkan menggunakan serial monitor pada PC.



Gambar 5. 2 Perancangan sistem node base

Pada gambar 5.2, dapat dijelaskan bahwa perancangan sistem node base pada penelitian ini yaitu menghubungkan mikrokontroler Arduino Nano dengan modul komunikasi wireless nRF24L01. Modul komunikasi wireless pada node base digunakan untuk menerima data sensor dari node client. Perancangan sistem dilakukan dengan menghubungkan pin-pin yang terdapat pada mikrokontroler Arduino Nano dengan pin-pin yang terdapat pada modul komunikasi wireless nRF24L01. Konfigurasi pin pada Arduino Nano dengan pin pada modul komunikasi wireless nRF24L01 adalah sebagai berikut :

Tabel 5. 1 Konfigurasi pin Arduino Nano dengan modul komunikasi wireless nRF24L01 pada node base

Arduino Nano	nRF24L01
Nama Pin	Nama Pin
Pin 1 GND	GND
Pin 2 VCC	3.3V
Pin 3 CE	D8
Pin 4 CS	D7

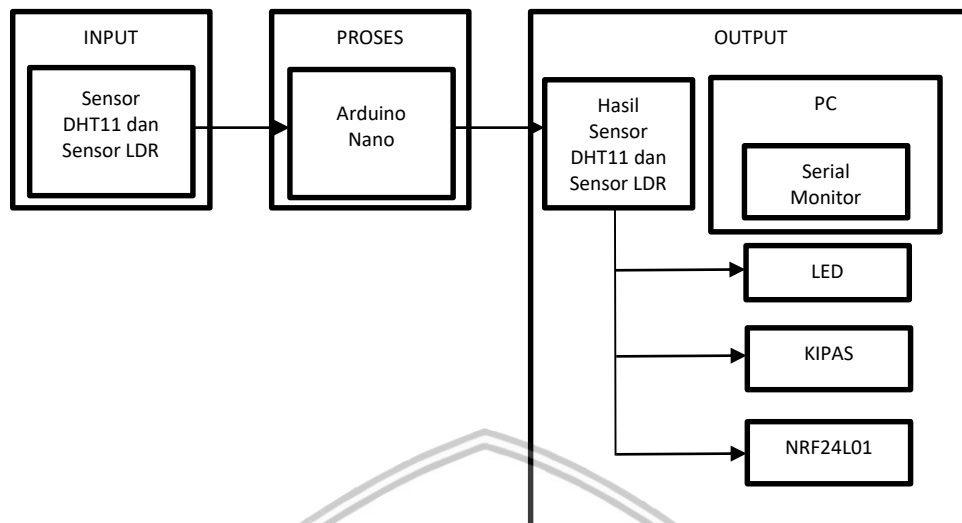
Pin 5 SCK	D13
Pin 6 MOSI	D11
Pin 7 MISO	D12

Tabel 5. 2 Konfigurasi fungsi dan prioritas *task* pada node *base*

Node Base		
Jenis Task	Fungsi	Prioritas
Task 1	Task yang berisikan tentang program yang menjalankan nRF24L01 untuk menerima data sensor dari node <i>client</i>	2
Task 2	Task yang berisikan tentang program yang mencetak data sensor dan menampilkannya melalui <i>serial monitor</i>	1

Berdasarkan Tabel 5.2, diketahui bahwa jenis *task* pada node *base* ada 2 yaitu *task 1* dan *task 2*. *Task 1* diberikan prioritas dengan nilai yang lebih tinggi dibandingkan *task 2* yaitu 2. *Task 1* diberikan prioritas dengan nilai yang lebih rendah dibandingkan nilai prioritas *task 1* yaitu 1. *Task 1* diberikan nilai prioritas yang lebih tinggi daripada *task 2* supaya *task* yang bekerja terlebih dahulu adalah *task 1*. Hal ini dikarenakan node *base* harus menerima data terlebih dahulu setelah itu dicetak dan ditampilkan melalui *serial monitor* sehingga data yang diterima dan ditampilkan dapat bersifat *real-time*.

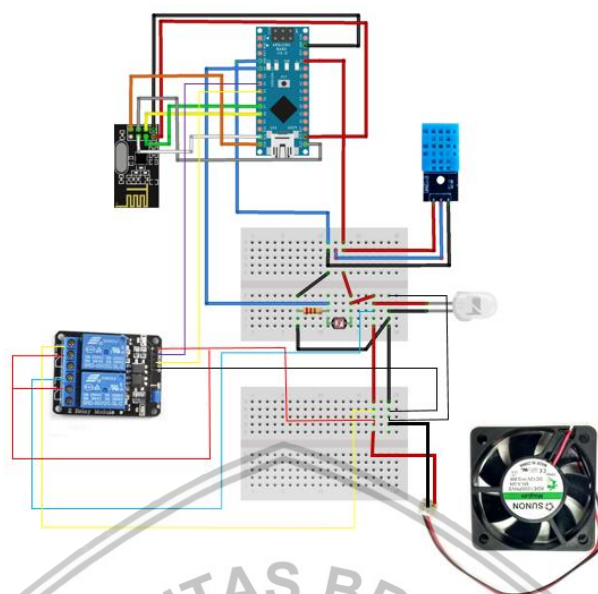
5.1.2.2 Perancangan Sistem Node *Client*



Gambar 5.3 Diagram blok node *client*

Pada gambar 5.3, dapat dijelaskan bahwa diagram blok pada node client memiliki :

1. Input data dilakukan oleh sensor DHT11 dan LDR.
2. Proses data dilakukan oleh Arduino Nano
3. Output data sensor ditampilkan menggunakan serial monitor pada PC.
4. Output dari data berupa suhu yang diambil oleh sensor DHT11 dapat menyalakan kipas jika diperoleh suhu lebih dari 25 derajat dan sebaliknya dapat menyalakan LED jika suhu kurang dari 25 derajat.
5. Hasil dari data sensor akan dikirimkan kepada node base melalui media komunikasi nRF24L01.



Gambar 5.4 Perancangan Sistem Node Client

Pada gambar 5.4, dapat dijelaskan bahwa perancangan sistem node client pada penelitian ini terdiri dari 5 bagian. Bagian yang pertama adalah perancangan mikrokontroler Arduino Nano dengan sensor DHT11. Perancangan sistem dilakukan dengan cara menghubungkan pin-pin yang ada pada mikrokontroler Arduino Nano dengan pin-pin yang ada pada sensor DHT11. Konfigurasi pin antara mikrokontroler Arduino Nano dengan sensor DHT11 adalah sebagai berikut :

Tabel 5. 3 Konfigurasi pin sensor DHT11 dengan mikrokontroler Arduino Nano pada node client

Sensor DHT11	Arduino Nano
Nama Pin	Nama Pin
VCC	5V
Signal	D2
GND	GND

Pada tabel 5.3, dapat dijelaskan bahwa konfigurasi pin antara sensor DHT11 dengan mikrokontroler Arduino Nano sebagai berikut : pin VCC pada sensor DHT11 terhubung dengan pin 5V pada Arduino Nano yang berarti sensor DHT11 menggunakan daya sebesar 5V. Pin Signal pada sensor DHT11 terhubung dengan pin D2 pada Arduino Nano yang digunakan sebagai input data sensing. Pin GND pada sensor DHT11 yang terhubung dengan pin GND pada Arduino Nano.

Bagian yang kedua yaitu perancangan mikrokontroler Arduino Nano dengan sensor LDR. Perancangan sistem dilakukan dengan menghubungkan pin-pin yang ada pada mikrokontroler Arduino Nano dengan pin-pin yang ada pada sensor LDR menggunakan jumper. Konfigurasi pin antara mikrokontroler Arduino Nano dengan sensor LDR adalah sebagai berikut :

Tabel 5. 4 Konfigurasi pin mikrokontroler Arduino Nano dengan sensor LDR pada node *client*

Arduino Nano	Sensor LDR
Nama Pin	Nama Pin
Pin GND	GND
Pin 5V	VCC

Pada tabel 5.3, dapat dijelaskan bahwa konfigurasi pin antara Arduino Nano dengan sensor LDR sebagai berikut : Pin GND pada mikrokontroler Arduino Nano terhubung dengan pin GND pada sensor LDR. Pin 5V pada Arduino Nano terhubung dengan pin VCC pada sensor LDR yang berarti sensor LDR menggunakan daya sebesar 5V.

Bagian ketiga yaitu perancangan mikrokontroler Arduino Nano dengan modul komunikasi wireless nRF24L01. Perancangan sistem dilakukan dengan menghubungkan pin-pin yang terdapat pada mikrokontroler Arduino Nano dengan pin-pin yang terdapat pada modul komunikasi wireless nRF24L01. Konfigurasi pin antara mikrokontroler Arduino Nano dengan modul komunikasi wireless nRF24L01 adalah sebagai berikut :

Tabel 5.5 Konfigurasi Pin mikrokontroler Arduino Nano dengan modul komunikasi *wireless* nRF24L01 pada node *client*.

Arduino Nano	nRF24L01
Nama Pin	Nama Pin
Pin 1 GND	GND
Pin 2 Vcc	3.3V
Pin 3 CE	D8
Pin 4 CS	D7
Pin 5 SCK	D13
Pin 6 MOSI	D11
Pin 7 MISO	D12

Bagian keempat yaitu perancangan mikrokontroler Arduino Nano dengan modul kipas. Perancangan sistem dilakukan dengan menghubungkan pin-pin yang terdapat pada mikrokontroler Arduino Nano dengan pin-pin yang terdapat pada modul kipas. Konfigurasi pin antara mikrokontroler Arduino Nano dengan modul kipas adalah sebagai berikut :

Tabel 5. 6 Konfigurasi Pin Relay dengan kipas

Kipas	Relay
Nama Pin	Nama Pin

VCC	OP2
GND	GND

Pada tabel 5.6, dapat dijelaskan bahwa konfigurasi pin antara Relay dengan modul kipas sebagai berikut : Pin OP2(*Output 2*) pada Relay terhubung dengan pin VCC pada kipas untuk mengatur kondisi tergantung nilai suhu yang didapat dari sensor DHT11. Pin GND pada Relay terhubung dengan pin GND pada kipas.

Bagian kelima yaitu perancangan Relay dengan LED. Perancangan sistem dilakukan dengan menghubungkan pin-pin yang terdapat pada Relay dengan pin-pin yang terdapat pada LED. Konfigurasi pin antara Relay dan LED adalah sebagai berikut :

Tabel 5. 7 Konfigurasi pin Relay dengan LED

LED	Relay
Nama Pin	Nama Pin
VCC	OP1
GND	GND

Pada tabel 5.7, dapat dijelaskan bahwa konfigurasi pin antara Relay dengan LED sebagai berikut : Pin OP1(*Output 1*) pada Relay terhubung dengan pin VCC pada LED untuk mengatur kondisi tergantung nilai suhu yang didapat dari sensor DHT11. Pin GND pada Relay terhubung dengan pin GND pada LED.

Bagian keenam yaitu perancangan Relay dengan Arduino Nano. Perancangan sistem dilakukan dengan menghubungkan pin-pin yang terdapat pada Relay dengan pin-pin yang terdapat pada Arduino Nano. Konfigurasi pin antara Relay dan Arduino Nano adalah sebagai berikut :

Tabel 5. 8 Konfigurasi pin Relay dengan Arduino Nano

Arduino Nano	Relay
Nama Pin	Nama Pin
VCC	VCC
GND	GND
D4	IN1
D5	IN2

Pada tabel 5.8, dapat dijelaskan bahwa konfigurasi pin antara Relay dengan LED sebagai berikut : Pin VCC pada Arduino Nano terhubung dengan pin VCC pada Relay. Pin Ground pada Arduino Nano terhubung dengan pin Ground pada Relay. Pin IN1(*Input 1*) pada Relay terhubung dengan pin D4 pada Arduino Nano untuk mengatur kondisi LED tergantung nilai suhu yang didapat dari sensor DHT11. Pin IN2(*Input 2*) pada Relay terhubung dengan pin D5 pada Arduino Nano untuk

mengatur kondisi LED tergantung nilai suhu yang didapat dari sensor DHT11. Pin GND pada Relay terhubung dengan pin GND pada LED.

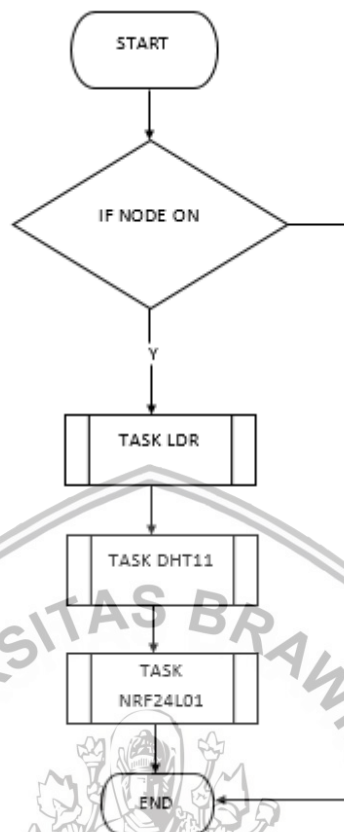
Tabel 5. 9 Konfigurasi prioritas dan fungsi task pada node *client*

Node <i>Client</i>		
Jenis Task	Fungsi	Prioritas
Task 1	Task yang berisikan tentang program yang menjalankan sensor LDR untuk <i>sensing</i> intensitas cahaya	3
Task 2	Task yang berisikan tentang program yang menjalankan sensor DHT11 untuk <i>sensing</i> suhu dan kelembaban	2
Task 3	Task yang berisikan tentang program nRF24L01 yang mengirimkan data sensor LDR dan DHT11 kepada node <i>base</i>	1

Berdasarkan Tabel 5.9, dapat diketahui bahwa jumlah *task* pada node *client* adalah 3. *Task* 1 memiliki nilai prioritas yang lebih tinggi dibanding *task* 2 dan *task* 3 yaitu 3. *Task* 2 memiliki nilai prioritas yang lebih tinggi dibanding *task* 3 dan lebih rendah dibanding *task* 1 yaitu 2. *Task* 3 memiliki nilai prioritas yang lebih rendah daripada *task* 1 dan *task* 2 yaitu 1. Pemberian prioritas pada masing-masing *task* yang berbeda disebabkan *task* yang berisi pengambilan data sensor yaitu *task* 1 dan *task* 2 harus berjalan terlebih dahulu daripada *task* 3. Hal ini supaya data sensor bisa didapatkan terlebih dahulu sebelum dikirimkan kepada node *base*. Dengan pemberian prioritas pada *task-task* seperti di atas maka pengiriman data sensor bersifat *real-time*.

5.1.3 Perancangan Metode *Preemptive Priority-Based Scheduling* pada Node *Client*

Node Client sebagai node yang berfungsi untuk mengakuisisi data sensor lalu dikirimkan kepada node base. Berikut ini adalah rancangan task-task yang ada pada Node Client menggunakan metode Preemptive Priority-Based Scheduling yang digambarkan melalui diagram alur :



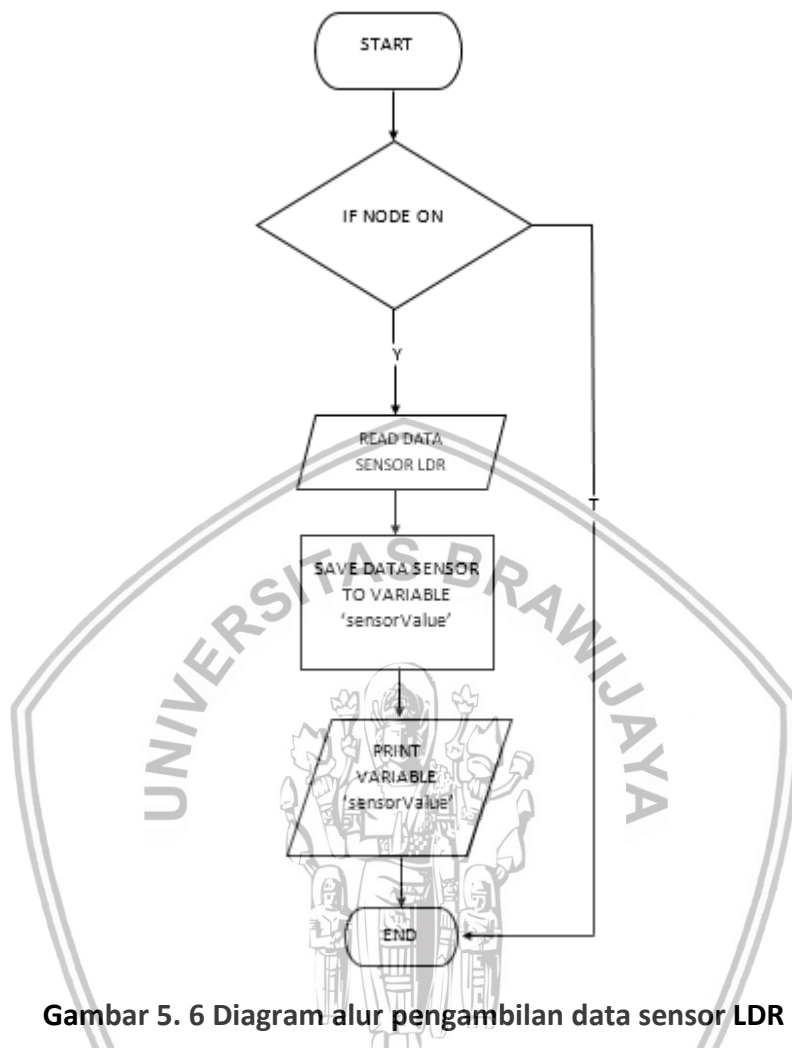
Gambar 5. 5 Diagram alur metode *Preemptive Priority-Based Scheduling* pada *Node Client*

Pada gambar 5.5, dapat dijelaskan bahwa proses penerapan metode Preemptive Priority-Based Scheduling dilakukan dengan inisialisasi task-task yang akan dijalankan. Task-task yang telah dibuat akan diberikan prioritas masing-masing supaya dapat berjalan secara bergantian. Task dengan prioritas tertinggi akan dijalankan terlebih dahulu dan dilanjutkan dengan task yang memiliki prioritas lebih rendah.

Setelah Node Client aktif maka task pengambilan data sensor LDR akan dijalankan terlebih dahulu karena memiliki prioritas tertinggi. Task kedua yang dijalankan adalah task pengambilan data sensor DHT11 karena memiliki prioritas lebih rendah dibandingkan dengan task pengambilan data sensor LDR. Task pengiriman data menggunakan media komunikasi wireless NRF24L01 dijalankan paling akhir karena memiliki prioritas terendah dibandingkan dengan task pengambilan data sensor LDR dan DHT11.

5.1.4 Perancangan *Task* pengambilan data sensor LDR pada *Node Client*

Node Client sebagai node yang berfungsi untuk mengakuisisi data sensor lalu dikirimkan kepada node base. Berikut adalah rancangan pengambilan data sensor LDR yang digambarkan melalui diagram alur :



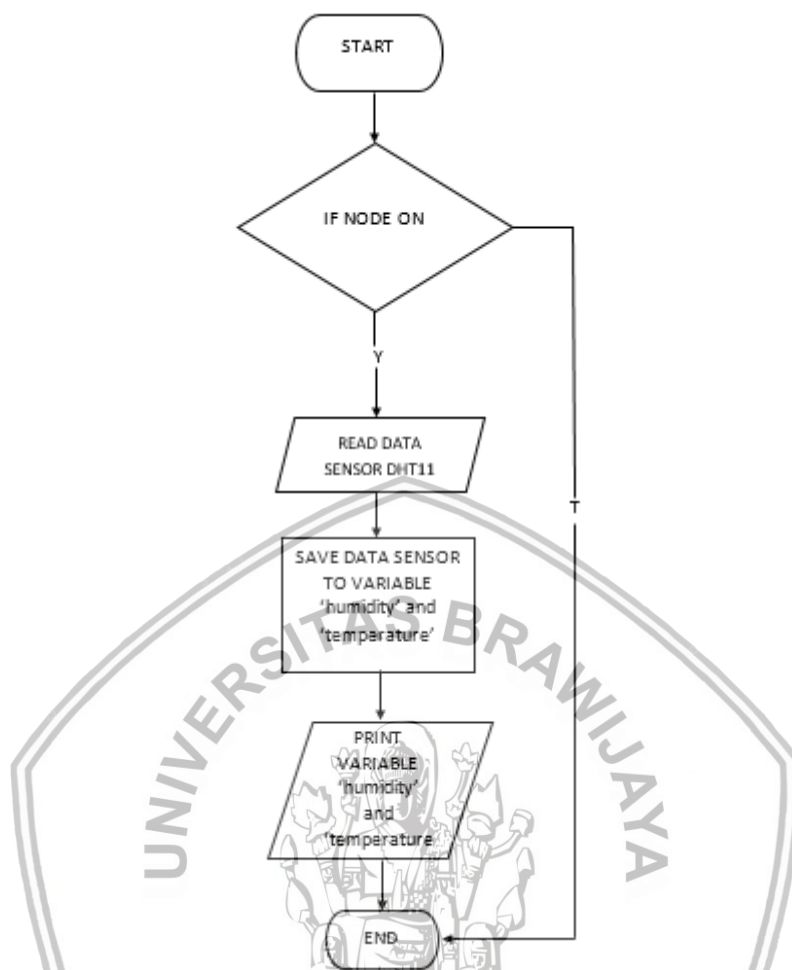
Gambar 5. 6 Diagram alur pengambilan data sensor LDR

Pada gambar 5.6, dapat dijelaskan bahwa proses pengambilan data sensor diawali dengan inisialisasi baudrate beserta pin sensor pada arduino nano untuk pengambilan data sensor. Inisialisasi baudrate untuk menentukan arduino nano bekerja sesuai dengan rate yang ditentukan. Inisialisasi pin sensor pada arduino nano untuk menentukan sensor bekerja pada pin yang telah ditetapkan.

Setelah Node Client aktif maka Node Client akan mulai mengambil data sensor LDR. Data yang telah didapatkan akan disimpan pada variabel sensorValue kemudian dicetak dan ditampilkan pada serial monitor menggunakan Arduino IDE.

5.1.5 Perancangan *Task* pengambilan data sensor DHT11 pada Node Client

Node Client sebagai node yang berfungsi untuk mengakuisisi data sensor lalu dikirimkan kepada node base. Berikut adalah rancangan pengambilan data sensor DHT11 yang digambarkan melalui diagram alur :



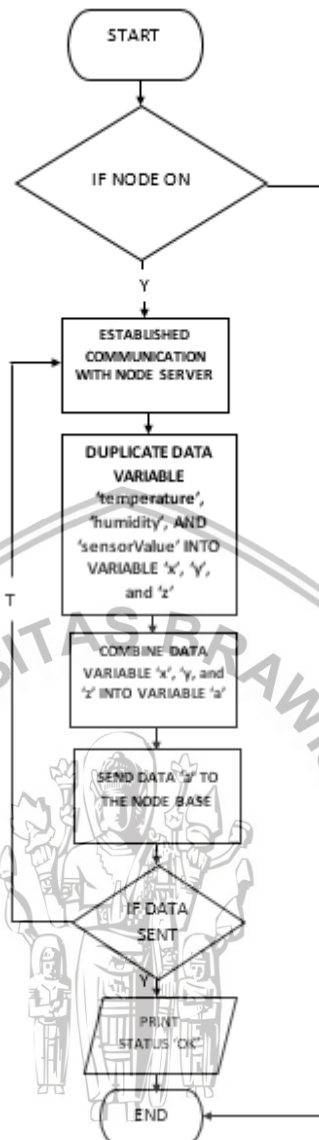
Gambar 5. 7 Diagram alur pengambilan data sensor DHT11

Pada gambar 5.7, dapat dijelaskan bahwa proses pengambilan data sensor diawali dengan inialisasi baudrate beserta pin sensor pada arduino nano untuk pengambilan data sensor DHT11. Inialisasi baudrate untuk menentukan arduino nano bekerja sesuai dengan rate yang ditentukan. Inialisasi pin sensor pada arduino nano untuk menentukan sensor bekerja pada pin yang telah ditetapkan.

Setelah Node Client aktif maka Node Client akan mulai mengambil data sensor DHT11. Data yang telah didapatkan akan disimpan pada variabel humidity dan temperature kemudian dicetak dan ditampilkan pada serial monitor menggunakan Arduino IDE.

5.1.6 Perancangan *Task* Pengiriman Data pada Node *Client*

Node Client sebagai node yang berfungsi untuk mengakuisisi data sensor lalu dikirimkan kepada node base. Berikut adalah rancangan pengiriman data sensor LDR dan DHT11 yang digambarkan melalui diagram alur :



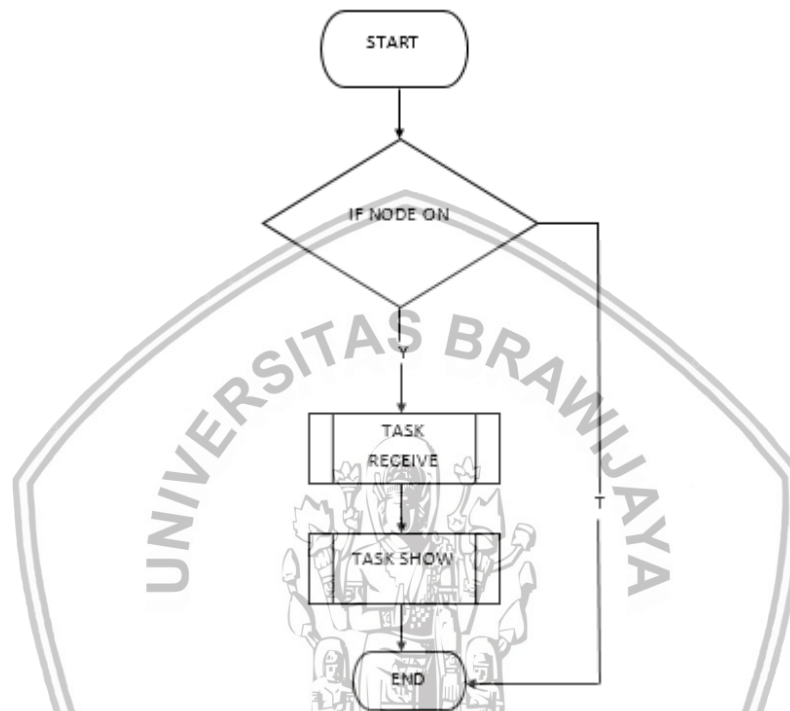
Gambar 5. 8 Diagram alur pengiriman data sensor pada node *Client*

Pada gambar 5.8, dapat dijelaskan bahwa proses pengiriman data sensor diawali dengan inialisasi baudrate beserta pin media komunikasi wireless NRF24L01 pada arduino nano. Inialisasi baudrate untuk menentukan arduino nano bekerja sesuai dengan rate yang ditentukan. Inialisasi pin NRF24L01 pada arduino nano untuk menentukan media komunikasi bekerja pada pin yang telah ditetapkan.

Setelah Node Client aktif maka Node Client akan mulai mengatur koneksi dengan node Base. Data sensor yang telah didapatkan dari sensor LDR dan dan DHT11 akan diduplikasi ke variabel x, y, dan z. Data-data yang telah diduplikasi tersebut akan dijadikan satu dengan dimasukkan ke dalam variabel a. Kemudian data variabel a akan dicetak dan ditampilkan pada serial monitor menggunakan arduino IDE.

5.1.7 Perancangan Metode *Preemptive Priority-Based Scheduling* pada Node Base

Node Base sebagai node yang berfungsi untuk menerima dan mencetak kemudian menampilkan data yang dikirim oleh node client. Berikut ini adalah rancangan task-task yang ada pada Node Base menggunakan metode Preemptive Priority-Based Scheduling yang digambarkan melalui diagram alur :



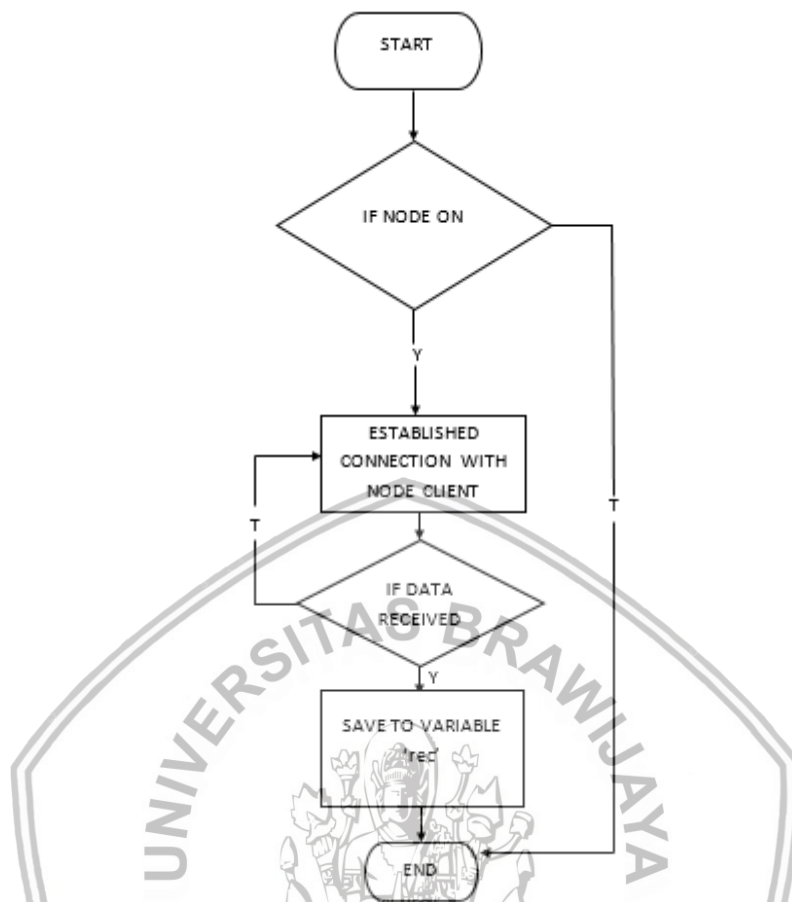
Gambar 5. 9 Diagram alur metode *preemptive priority based-scheduling* pada node base

Pada gambar 5.9, dapat dijelaskan bahwa proses penerapan metode Preemptive Priority-Based Scheduling dilakukan dengan inisialisasi task-task yang akan dijalankan. Task-task yang telah dibuat akan diberikan prioritas masing-masing supaya dapat berjalan secara bergantian. Task dengan prioritas tertinggi akan dijalankan terlebih dahulu dan dilanjutkan dengan task yang memiliki prioritas lebih rendah.

Setelah Node Base aktif maka task penerimaan data sensor menggunakan media komunikasi wireless NRF24L01 akan dijalankan terlebih dahulu karena memiliki prioritas tertinggi. Task kedua yang dijalankan adalah task pencetakan data sensor karena memiliki prioritas yang lebih rendah.

5.1.8 Perancangan *Task* Penerimaan Data pada Node Base

Node Base sebagai node yang berfungsi untuk menerima dan menampilkan data yang dikirim oleh node client. Berikut ini adalah rancangan penerimaan data pada node base yang digambarkan melalui diagram alur :



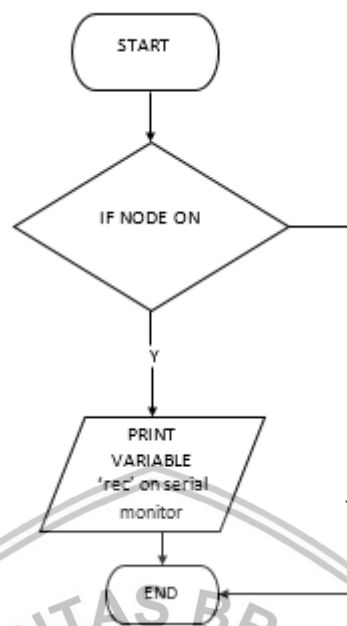
Gambar 5. 10 Diagram alur penerimaan data pada node *base*

Pada gambar 5.10, dapat dijelaskan bahwa proses penerimaan data sensor pada node base diawali dengan inisialisasi baudrate beserta pin media komunikasi wireless NRF24L01 pada arduino nano. Inisialisasi baudrate untuk menentukan arduino nano bekerja sesuai dengan rate yang ditentukan. Inisialisasi pin NRF24L01 pada arduino nano untuk menentukan media komunikasi bekerja pada pin yang telah ditetapkan.

Setelah Node Base aktif maka Node Base akan mulai mengatur koneksi dengan node Client. Data sensor yang telah diterima akan disimpan pada variabel *rec*.

5.1.9 Perancangan *Task* pencetakan data sensor yang telah diterima pada Node *Base*

Node Base sebagai node yang berfungsi untuk menerima dan mencetak kemudian menampilkan data yang dikirim oleh node client. Berikut ini adalah rancangan pencetakan data sensor yang telah diterima yang digambarkan melalui diagram alur :



Gambar 5. 11 Diagram alur pencetakan data sensor pada node *base*

Pada gambar 5.11, dapat dijelaskan bahwa proses pencetakan data sensor pada node base diawali dengan inisialisasi baudrate beserta pin media komunikasi wireless NRF24L01 pada arduino nano. Inisialisasi baudrate untuk menentukan arduino nano bekerja sesuai dengan rate yang ditentukan.

Setelah Node Base aktif maka Node Base akan mulai mencetak data sensor yang telah diterima yaitu variabel *rec* lalu ditampilkan pada serial monitor menggunakan Arduino IDE.

5.2 Implementasi

Pada tahap ini akan dijelaskan tahap implementasi dari semua perancangan penelitian yang telah dilakukan. Pada tahap ini membahas langkah-langkah penelitian dalam mengimplementasikan sistem operasi real-time menggunakan metode Preemptive Priority-Based Scheduling sebagai metode untuk mengatasi masalah pengiriman dan penerimaan data pada node client dan node base.

5.2.1 Spesifikasi *Hardware*

Perancangan penelitian “Implementasi Sistem Operasi Real-Time pada Arduino Nano dengan media Komunikasi NRF24L01 Untuk Pengukuran Suhu dan Kelembaban” ini dalam menerapkan sistem operasi yang bersifat real-time menggunakan beberapa komponen hardware, diantaranya sensor DHT11, sensor LDR, Arduino Nano, dan modul komunikasi wireless nRF24L01. Spesifikasi dari komponen-komponen tersebut adalah :

- Sensor DHT11

Sensor DHT11 adalah sebuah sensor digital yang digunakan untuk mengukur suhu dan kelembaban di sekelilingnya.

- Sensor LDR

Sensor LDR adalah sebuah sensor analog yang digunakan untuk menangkap intensitas cahaya yang berada disekitarnya.

- Perangkat Mikrokontroler

Mikrokontroler yang digunakan dalam penelitian ini adalah Arduino Nano.

- Modul komunikasi *wireless* nRF24L01

Modul *wireless* nRF24L01 adalah modul yang digunakan untuk melakukan komunikasi antara node *client* dan node *base*.

- LED

LED adalah modul yang digunakan sebagai output pada node *client*.

- Kipas

Kipas adalah modul yang digunakan untuk melakukan komunikasi antara node *client*.

- Laptop/ Komputer

Laptop digunakan untuk membuat sekaligus mengupload program ke mikrokontroler. Spesifikasi laptop yang digunakan pada penelitian ini adalah sebagai berikut :

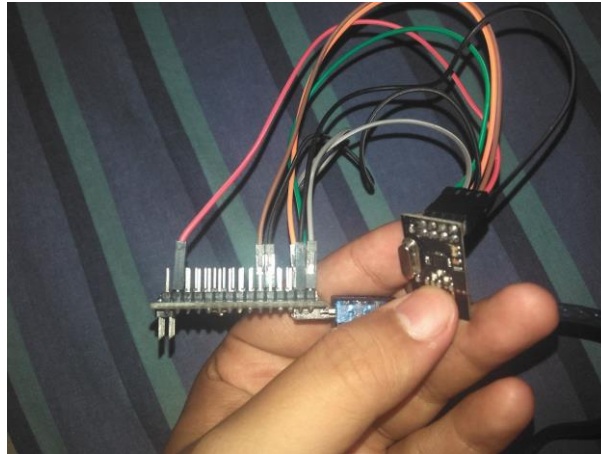
Tabel 5. 10 Spesifikasi Laptop

Laptop Asus X555O	
Processors	AMD Quad Core A10-9600P
Memory (RAM)	8GB
Hard Disk	1TB

Pada tabel 5.10, dapat dijelaskan bahwa Laptop yang digunakan adalah Asus bertipe X555O yang memiliki Prosesor AMD Quad Core A10-9600P, RAM 8GB, dan kapasitas Hard Disk 1TB.

5.2.2 Implementasi Komunikasi *Hardware*

Pada masing-masing node, implementasi komunikasi hardware dilakukan secara serial dengan menghubungkan board Arduino ke Laptop untuk pemrograman dan pengujian. Pada node base dan node client, untuk proses komunikasi atau pengiriman data akan dilakukan secara wireless dengan menggunakan modul komunikasi wireless nRF24L01.



Gambar 5.12 Implementasi nRF24L01 sebagai Komunikasi *Wireless*

Pada gambar 5.12, dapat dijelaskan bahwa saat sensor LDR dan sensor DHT11 yang terdapat dalam rancangan sistem node client telah melakukan sensing data, maka data hasil sensing tersebut akan dikirimkan kepada node base dan ditampilkan melalui serial monitor.

5.2.3 Implementasi *Hardware*

Pada penelitian ini, implementasi hardware adalah pemasangan seluruh komponen-komponen yang digunakan. Terdapat 2 pengelompokan hardware, yaitu : hardware pada Node base dan hardware pada node client.



Gambar 5.13 Komponen *Hardware*

Pada gambar 5.13, dapat dijelaskan bahwa pemasangan hardware pada node base meliputi Arduino Nano untuk menampilkan data hasil deteksi sensor dan receiver modul komunikasi wireless nRF24L01 yang digunakan untuk menerima data. Pada node client, pemasangan hardware meliputi sensor LDR sebagai perangkat penangkap intensitas cahaya, sensor DHT11 sebagai perangkat pengukur suhu dan kelembaban, Arduino Nano sebagai pengolah data, dan

transmitter modul komunikasi wireless nRF24L01 sebagai perangkat pengirim data.

Implementasi hardware pada node base dilakukan dengan menghubungkan nRF24L01 dengan Arduino Nano. Implementasi hardware pada node dilakukan dengan menghubungkan sensor LDR, sensor DHT11 dan modul komunikasi wireless nRF24L01 dengan mikrokontroler Arduino Nano.

5.2.4 Implementasi *Software*

Pada penelitian ini diperlukan library pada sisi pemrograman software agar sistem dapat bekerja sesuai dengan fungsinya masing-masing. Fungsi library sendiri untuk menunjang kinerja atau fungsionalitas dari sistem yang telah dirancang. Fungsi-fungsi yang ada dengan terdapatnya library dalam penelitian ini adalah sistem mampu menerapkan sistem operasi real-time menggunakan metode Preemptive Priority-Based Scheduling, mampu mendapatkan nilai sensor yang berupa data cahaya, suhu dan kelembaban, dan mampu mengirimkan nilai data sensor tersebut. Library akan dimasukkan ke dalam program yang tertanam pada mikrokontroler, yaitu Arduino Nano. Implementasi library code pada sisi node client dapat dilihat pada Tabel 5.8 sedangkan implementasi library code pada sisi node base dapat dilihat pada Tabel 5.9.

Tabel 5. 11 Library code node client

1	#include <Arduino_FreeRTOS.h>
2	#include "DHT.h"
3	#include <SPI.h>
4	#include <Mirf.h>
5	#include <nRF24L01.h>
6	#include <MirfHardwareSpiDriver.h>

Tabel 5. 12 Library code node base

1	#include <Arduino_FreeRTOS.h>
2	#include <SPI.h>
3	#include <Mirf.h>
4	#include <nRF24L01.h>
5	#include <MirfHardwareSpiDriver.h>

5.2.4.1 Implementasi Metode *Preemptive Priority-Based Scheduling* Pada Node Client

Pada penelitian ini diimplementasikan metode Preemptive Priority-Based Scheduling pada node client maupun node base. Metode ini diimplementasikan supaya tidak terjadi tabrakan data (collision data) dengan cara melakukan

penjadwalan(scheduling) pada masing-masing task. Penjadwalan(Scheduling) pada masing-masing task dapat dilakukan dengan memberikan prioritas pada masing-masing task. Dengan diberikannya prioritas pada masing-masing task, maka task dengan prioritas lebih tinggi akan dijalankan terlebih dahulu kemudian dilanjutkan dengan prioritas yang lebih rendah. Hal ini bertujuan agar data-data yang diolah oleh mikrokontroler tidak saling bertabrakan. Tahap pertama yang diperlukan dalam menerapkan metode ini adalah inisialisasi task-task yang akan dijalankan. Tahap ini dilakukan supaya sistem mengetahui task mana saja yang harus dijalankan. Implementasi kode program inisialisasi task dapat dilihat pada Tabel 5.10 untuk node client, sedangkan Tabel 5.11 untuk node base.

Tabel 5. 13 Kode program inisialisasi *task* pada node *client*

1	void TaskBlink(void *pvParameters);
2	void TaskDigitalRead(void *pvParameters);
3	void TaskLDR(void *pvParameters);

Tabel 5. 14 Kode program inisialisasi *task* pada node *base*

1	void TaskReceive(void *pvParameters);
2	void TaskShow(void *pvParameters);

Tahap selanjutnya adalah dengan memberikan prioritas yang berbeda pada masing-masing task. Tahap ini dilakukan supaya sistem dapat menjalankan task-task yang ada secara berurutan. Implementasi kode program penentuan prioritas pada masing-masing task dapat dilihat pada Tabel 5.12 untuk node *client* sedangkan Tabel 5.13 untuk node *base*. Pada node *client*, *task* yang pertama dijalankan adalah *TaskLDR* untuk membaca data sensor intensitas cahaya dengan diberikan nilai 3. *Task* kedua yang dijalankan adalah *TaskDigitalRead* dimana *task* ini menjalankan program sensor DHT11 untuk membaca data sensor suhu dan kelembaban dengan diberikan nilai 2. *Task* ketiga yang dijalankan adalah *TaskBlink* dimana *task* ini menjalankan program untuk mengirimkan data sensor LDR dan DHT11 kepada node *base* dengan diberikan nilai 1. Sedangkan pada sensor node *base* memiliki 2 *task* yang berisi *task* yang menerima data sensor dan *task* yang menampilkan data sensor pada *serial monitor*. Pada node *base*, *task* yang pertama dijalankan adalah *TaskReceive* untuk menerima data sensor DHT11 dan LDR yang dikirimkan oleh node *client* dengan diberikan nilai 2. *Task* kedua yang dijalankan adalah *TaskShow* untuk menampilkan data sensor DHT11 dan LDR melalui *serial monitor* dengan diberikan nilai 1.

Tabel 5. 15 Kode program pemberian prioritas pada node *client*

1	void setup() {
2	
3	// Now set up two tasks to run independently.


```

4  xTaskCreate(
5      TaskBlink
6      , (const portCHAR *)"Blink"  // A name just for humans
7      , 128 // Stack size
8      , NULL
9      , 1
10     , NULL );
11
12  xTaskCreate(
13      TaskDigitalRead
14      , (const portCHAR *) "DigitalRead"
15      , 128 // This stack size can be checked & adjusted by reading Highwater
16      , NULL
17      , 2
18      , NULL );
19
20  xTaskCreate(
21      TaskLDR
22      , (const portCHAR *) "AnalogRead"
23      , 128 // This stack size can be checked & adjusted by reading Highwater
24      , NULL
25      , 3
26      , NULL );
27  }
28  void loop()
29  {
30      // Empty. Things are done in Tasks.
31  }

```

Tabel 5. 16 Kode program pemberian prioritas pada node *base*

```

1  void setup() {
2      xTaskCreate(
3          TaskReceive

```

4	, (const portCHAR *)"DigitalRead"
5	, 128
6	, NULL
7	, 2
8	, NULL);
9	
10	xTaskCreate(
11	TaskShow
12	, (const portCHAR *)"Blink"
13	, 128
14	, NULL
15	, 1
16	, NULL);
17	}
18	void loop() {
19	}

5.2.4.2 Implementasi *Task* Pengambilan Data Sensor LDR Pada Node *Client*

Setelah masing-masing task diberikan prioritas, maka tahap selanjutnya adalah menentukan tugas dari masing-masing task. Pada task ini bertugas untuk melakukan pengambilan data sensor cahaya menggunakan sensor LDR. Task ini telah diberi prioritas pertama dengan nilai 3 sehingga sistem akan menjalankan task pengambilan data sensor LDR terlebih dahulu. Implementasi kode program task pengambilan data sensor LDR dapat dilihat pada Tabel 5.14.

Tabel 5. 17 Kode program *task* sensor LDR pada node *client*

1	Void TaskLDR(void *pvParameters) // This is a task.
2	{
3	(void) pvParameters;
4	Serial.begin(9600);
5	for (;;)
6	{ sensorValue = analogRead(sensorPin);
7	Serial.print("\t\tIni Task 1\n");
8	Serial.print("SensorLDR :");
9	Serial.print(sensorValue);

```

10 Serial.print("\n\n");
11 vTaskDelay(1500 / portTICK_PERIOD_MS);
12 }
13 }

```

5.2.4.3 Implementasi Task Pengambilan Data Sensor DHT11 Pada Node Client

Task ini bertugas untuk melakukan pengambilan data sensor suhu dan kelembaban menggunakan sensor DHT11. Task ini diberikan prioritas kedua dengan nilai 2 sehingga sistem akan menjalankan task ini setelah task pengambilan data sensor LDR. Implementasi kode program task pengambilan data sensor DHT11 dapat dilihat pada Tabel .

Tabel 5. 18 Kode program task sensor DHT11 pada node client

```

1 void TaskDigitalRead(void *pvParameters) // This is a task.
2 {
3     (void) pvParameters;
4     Serial.begin(9600);
5     dht.setup(2);
6     for (;;)
7     {
8         Serial.println("\t\tIni Task 2\n");
9         Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)");
10        delay(dht.getMinimumSamplingPeriod());
11        humidity = dht.getHumidity();
12        temperature = dht.getTemperature();
13        Serial.print(dht.getStatusString());
14        Serial.print("\t");
15        Serial.print(humidity, 1);
16        Serial.print("\t\t");
17        Serial.print(temperature, 1);
18        Serial.print("\t\t");
19        Serial.println(dht.toFahrenheit(temperature), 1);
20        Serial.print("\n\n");
21        vTaskDelay(1500 / portTICK_PERIOD_MS);
22    }

```

23	}
----	---

5.2.4.4 Implementasi *Task* Pengiriman Data Sensor Pada Node *Client*

Task ini bertugas untuk melakukan pengiriman data sensor cahaya, suhu dan kelembaban yang telah dilakukan oleh task pengambilan data sensor LDR dan sensor DHT11. Pengiriman data sensor menggunakan media komunikasi wireless NRF24L01. Task ini diberikan prioritas ketiga dengan nilai 1 sehingga sistem akan menjalankan task ini setelah task pengambilan data sensor LDR dan DHT11. Implementasi kode program task pengiriman data sensor dapat dilihat pada Tabel 5.16.

Tabel 5. 19 Kode program *task* pengiriman data sensor pada node *client*

1	Void TaskBlink(void *pvParameters) // This is a task.
2	{
3	(void) pvParameters;
4	Serial.begin(9600);
5	Mirf.spi = &MirfHardwareSpi;
6	Mirf.init();
7	Mirf.setTADDR((byte *)"serv1");
8	Mirf.payload = 32;
9	Mirf.config();
10	for (;;) // A Task shall never return or exit.
11	{
12	{
13	Serial.println("Beginning ... ");
14	char x[4], y[4], z[4];
15	sprintf(x, "%d", temperature);
16	sprintf(y, "%d", humidity);
17	sprintf(z, "%d", sensorValue);
18	strcpy(a, x);
19	strcat(a, y);
20	strcat(a, z);
21	Mirf.send((byte *)&a);
22	while (Mirf.isSending()) { }
23	Serial.print("\n");

24	Serial.print("\t\tIni Task 3\n");
25	Serial.print("\t Finished sending \t");
26	Serial.print("\n");
27	Serial.print("\ttemperature :\t");
28	Serial.print(a[0]);
29	Serial.print(a[1]);
30	Serial.print("\n");
31	Serial.print("\thumidity :\t");
32	Serial.print(a[2]);
33	Serial.print(a[3]);
34	Serial.print("\n");
35	Serial.print("\tLDR :\t");
36	Serial.print(a[4]);
37	Serial.print(a[5]);
38	Serial.print("\n\n");
39	vTaskDelay(1500 / portTICK_PERIOD_MS);
40	}
41	}

5.2.4.5 Implementasi Task Penerimaan Data Sensor Pada Node Base

Task ini bertugas untuk melakukan penerimaan data sensor cahaya, suhu dan kelembaban yang telah dikirim oleh node client. Penerimaan data sensor menggunakan media komunikasi wireless NRF24L01. Task ini diberikan prioritas tertinggi sehingga sistem akan menjalankan task ini terlebih dahulu. Implementasi kode program task penerimaan data sensor dapat dilihat pada Tabel 5.17.

Tabel 5. 20 Kode program task penerimaan data sensor pada node base

1	void TaskReceive(void *pvParameters) {
2	(void) pvParameters;
3	Serial.begin(9600);
4	Mirf.spi = &MirfHardwareSpi;
5	Mirf.init();
6	Mirf.setRADDR((byte *)"serv1");
7	Mirf.payload = 32;
8	Mirf.config();

9	for (;;) {
10	if (!Mirf.isSending() && Mirf.dataReady()) {
11	Mirf.getData((byte *)&rec);
12	Serial.println("Ini Task 1");
13	vTaskDelay(500 / portTICK_PERIOD_MS);
14	}
15	}
16	}

5.2.4.6 Implementasi Pencetakan Data Sensor Pada Node *Base*

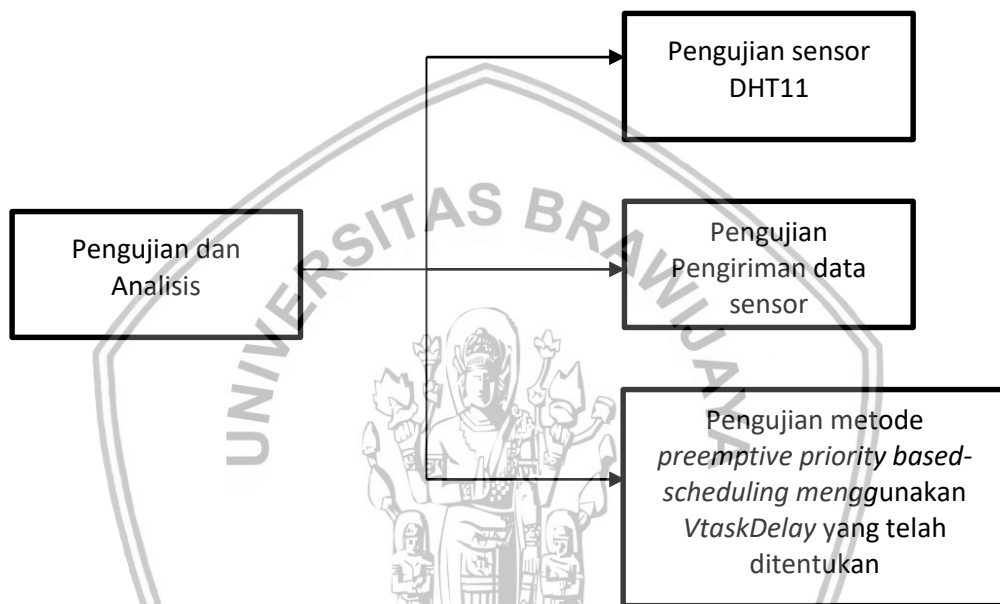
Task ini bertugas untuk melakukan pencetakan dan menampilkan data sensor pada serial monitor menggunakan Arduino IDE. Task ini berada di node base. Task ini diberikan prioritas kedua sehingga sistem akan menjalankan task ini setelah task penerimaan data sensor. Implementasi kode program task pencetakan data sensor dapat dilihat pada Tabel 5.18.

Tabel 5. 21 Kode program *task* pencetakan data sensor pada node *base*

1	void TaskShow (void *pvParameters) {
2	(void) pvParameters;
3	Serial.begin(9600);
4	for (;;) {
5	Serial.println("\t\t Ini Task 2");
6	Serial.println("Listening...");
7	Serial.print("Kelembaban : ");
8	Serial.print(rec[0]); Serial.print(rec[1]);
9	Serial.print("\t suhu : ");
10	Serial.print(rec[2]); Serial.println(rec[3]);
11	Serial.print("\t sensor LDR : ");
12	Serial.print(rec[4]); Serial.println(rec[5]);
13	vTaskDelay(500 / portTICK_PERIOD_MS);
14	}}

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini memuat proses pengujian sistem multi-sensor dan ditambahkan media komunikasi wireless nRF24L01 dengan mengimplementasikan RTOS. Terdapat 3 pengujian dalam sistem ini, yaitu waktu yang dibutuhkan dalam melakukan eksekusi program tiap *task*, pengujian pengiriman dan penerimaan data sensor menggunakan media komunikasi nRF24L01 dan pengujian perbedaan sistem yang menggunakan RTOS dengan sistem tanpa menggunakan RTOS. Hal tersebut ditunjukkan pada Gambar 6.1.



Gambar 6. 1 Grafik pengujian dan analisis

6.1 Pengujian Sensor DHT11

6.1.1 Tujuan

Untuk mengetahui kinerja sensor DHT11 dan LDR yang digunakan apakah berfungsi dengan baik atau tidak. Pengukuran hasil dari sensor DHT11 apakah sama dengan pengukuran suhu yang digunakan sehari-hari yaitu termometer ruangan.

6.1.2 Prosedur

Untuk menguji kinerja sensor DHT11 dalam pengukuran suhu, maka suhu yang didapat akan dibandingkan dengan hasil pengukuran dari termometer. Sensor akan dibandingkan dengan hasil pengukuran dari termometer ruangan. Output dari data suhu pada sensor dapat menyalakan kipas ataupun led bergantung pada nilai suhu sensor. Pengujian menggunakan program yang sudah diimplementasikan Real Time Operating System. Prosedur pengujian kinerja sensor DHT11 meliputi :

1. Sambungkan alat dengan komputer atau laptop
2. Buka *source code* yang dibuat untuk mengukur suhu menggunakan sensor DHT11. Setelah itu *compile* dan *upload source code* programnya.
3. Amati dan kumpulkan hasil data sensor DHT11 melalui *serial monitor* kemudian hitung perbandingan hasil yang didapat dengan termometer ruangan. Hasil perhitungan persentase error sensor dengan persamaan

$$\%Error = \frac{\text{Nilai Sensor} - \text{Nilai Termometer}}{\text{Nilai Termometer}} \times 100\% \quad (6.1)$$

4. Kesimpulan.

6.1.3 Hasil dan Analisis

Hasil dari pengujian suhu dari sensor DHT11 yang digunakan dalam sistem beserta perhitungan hasil error dengan nilai suhu pada termometer ruangan ditampilkan pada Tabel 6.1, output dari sistem untuk kondisi pada kipas ditampilkan pada Tabel 6.2, sedangkan output dari sistem untuk kondisi pada LED ditampilkan pada Tabel 6.3.

Tabel 6. 1 Hasil perbandingan sensor DHT11 dengan termometer ruangan

Suhu pada DHT11	Suhu pada Termometer Ruangan	Persentase Error
27	27	0
27	27	0
27	27	0
27	27	0
27	27	0
27	27	0
27	27	0
27	27	0
27	27	0
27	27	0

Berdasarkan analisis Tabel 6.1, rata-rata error pada sensor adalah 0%. Percobaan dilakukan sebanyak 10 kali. Dengan percobaan sebanyak 10 kali dapat disimpulkan bahwa kinerja sensor sangat baik. Jika sensor memiliki persentase error lebih dari 0% maka dapat mempengaruhi kinerja dari sistem di bagian output sistem. Suhu pada pengujian bernilai sama karena data suhu didapatkan dalam satu waktu.

Tabel 6. 2 Output kondisi kipas

Suhu pada DHT11	Kipas	LED
27	Nyala	Mati
27	Nyala	Mati

27	Nyala	Mati
27	Nyala	Mati
27	Nyala	Mati
27	Nyala	Mati
27	Nyala	Mati
27	Nyala	Mati
27	Nyala	Mati
27	Nyala	Mati

Berdasarkan analisis tabel 6.2, kipas akan menyala ketika suhu berada pada 27 derajat celcius sedangkan LED dalam keadaan mati. Keadaan ini sudah sesuai dengan sistem yang diatur karena kipas akan menyala ketika berada di atas suhu 25 derajat celcius dan LED akan mati begitupun sebaliknya.

Tabel 6. 3 Output kondisi LED

Suhu pada DHT11	Kipas	LED
24	Mati	Nyala
24	Mati	Nyala
24	Mati	Nyala
24	Mati	Nyala
24	Mati	Nyala
24	Mati	Nyala
24	Mati	Nyala
24	Mati	Nyala
24	Mati	Nyala
24	Mati	Nyala

Berdasarkan analisis Tabel 6.3, LED akan menyala ketika suhu berada pada 24 derajat celcius sedangkan kipas dalam keadaan mati. Keadaan ini sudah sesuai dengan sistem yang diatur karena LED akan menyala ketika suhu berada di bawah 25 derajat celcius sedangkan kipas akan mati begitupun sebaliknya.

6.2 Pengujian Media Komunikasi Wireless nRF24L01

6.2.1 Tujuan

Untuk mengetahui kinerja media komunikasi *wireless* nRF24L01 yang digunakan apakah berfungsi dengan baik atau tidak. Pengujian data sensor yang dikirimkan oleh node client apakah sudah sesuai dengan data yang diterima oleh node base.

6.2.2 Prosedur

Untuk menguji kinerja media komunikasi *wireless* nRF24L01 untuk pengiriman dan penerimaan data sensor, maka data sensor yang dikirim melalui node *client* akan dibandingkan dengan hasil data sensor yang diterima oleh node *base*.

Pengujian menggunakan program yang sudah diimplementasikan Real Time Operating System. Prosedur pengujian kinerja nRF24L01 meliputi :

1. Sambungkan alat dengan komputer atau laptop
2. Buka *source code* yang dibuat untuk pengiriman dan penerimaan data sensor. Setelah itu *compile* dan *upload source code* programnya.
3. Amati dan kumpulkan hasil data sensor yang dikirim dan diterima melalui *serial monitor* pada masing-masing node kemudian hitung perbandingan hasil yang didapat. Hasil perhitungan persentase error sensor dengan persamaan

$$\%Error = \frac{\text{Data sensor yang dikirim} - \text{Data sensor yang di terima}}{\text{Data sensor yang dikirim}} \times 100 \quad (6.2)$$

4. Kesimpulan.

6.2.3 Hasil dan Analisis

Tabel 6. 4 Analisis hasil pengujian pengiriman dan penerimaan data sensor

Percobaan ke-	Node Client			Node Base			Persentase Error
	Pengiriman data sensor			Penerimaan data sensor			
	LDR	DHT11		LDR	DHT11		
		Suhu (°C)	Kelembaban (RH)		Suhu (°C)	Kelembaban (RH)	
1	88	27	77	88	27	77	0
2	94	27	77	94	27	77	0
3	94	27	77	94	27	77	0
4	92	27	77	92	27	77	0
5	92	27	77	92	27	77	0
6	93	27	77	93	27	77	0
7	94	27	77	94	27	77	0
8	93	27	77	93	27	77	0
9	93	27	77	93	27	77	0
10	93	27	77	93	27	77	0

Dari hasil uji yang didapat dari node client sebagai pengirim dan node base sebagai penerima dapat disimpulkan bahwa data sensor yang dikirimkan dan diterima sama dengan persentasi error 0%. Hal ini mengindikasikan bahwa media komunikasi wireless nRF24L01 dapat bekerja dengan baik meskipun data sensor yang dikirimkan lebih dari satu sensor.

6.3 Pengujian Metode *Preemptive Priority Based-Scheduling*

6.3.1 Tujuan

Untuk mengetahui kinerja metode preemptive priority based-scheduling yang digunakan apakah berfungsi dengan baik atau tidak. Pengujian eksekusi task pada masing-masing node apakah sudah berjalan sesuai metode preemptive priority based-scheduling atau tidak.

6.3.2 Prosedur

Untuk menguji kinerja metode *Preemptive Priority Based-Scheduling* sudah berjalan dengan semestinya pada masing-masing task, maka sistem sudah diatur dengan *vTaskDelay* selama 3000 ms per-periode kemudian diuji waktu eksekusi setiap tasknya. Pada node *client* terdapat 3 *task* yang berbeda. Masing-masing *task* diberikan prioritas yang berbeda juga. *Task* LDR yaitu *task* yang berisikan program sensor LDR diberikan prioritas paling tinggi dengan nilai 3, *task* DHT11 yaitu *task* yang berisikan program sensor DHT11 memiliki prioritas kedua dengan diberikan nilai 2, dan *task* NRF24L01 yaitu *task* yang berisikan program pengiriman data sensor menggunakan NRF24L01 memiliki prioritas terendah dengan diberikan nilai 1. Untuk membantu perhitungan waktu eksekusi tiap *task* digunakan fungsi timer pada Arduino yaitu fungsi “*micros()*” dan “*milis()*”. Fungsi “*micros()*” untuk menghitung waktu dalam satuan mikro detik dan fungsi “*milis()*” untuk menghitung waktu dalam satuan mili detik. Pengujian menggunakan program yang sudah diimplementasikan Real Time Operating System. Prosedur pengujian kinerja metode *Preemptive Priority Based-Scheduling* meliputi :

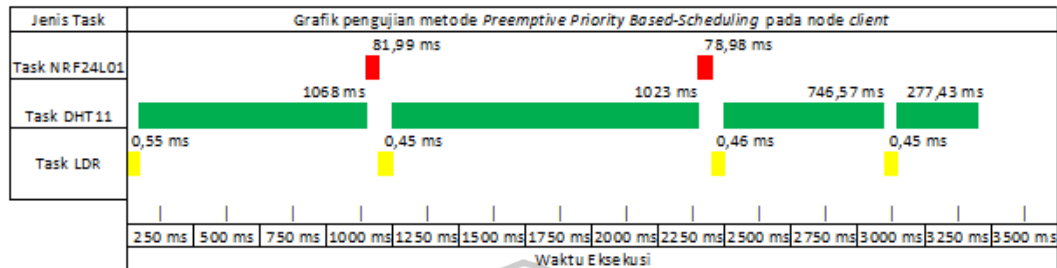
1. Sambungkan alat dengan komputer atau laptop
2. Buka *source code* yang telah diimplementasikan . Setelah itu *compile* dan *upload source code* programnya.
3. Amati dan kumpulkan hasil data eksekusi *task* pada masing-masing node kemudian hitung hasil eksekusi waktu yang didapat sesuai *vTaskDelay* yang ditentukan.
4. Kesimpulan.

6.3.3 Hasil dan Analisis

Tabel 6. 5 Analisis hasil pengujian metode *Preemptive Priority Based-Scheduling*

Node Client		
Percobaan Ke	Jenis Task	Waktu Eksekusi
1	Task LDR	0,55 ms
2	Task DHT11	1068 ms
3	Task NRF24L01	81,99 ms
4	Task LDR	0,45 ms
5	Task DHT11	1023 ms
6	Task NRF24L01	78,98 ms

7	Task LDR	0,46 ms
8	Task DHT11	1024 ms
9	Task LDR	0,45 ms
10	Task DHT11	277,43 ms



Gambar 6. 2 Analisis hasil pengujian metode *Preemptive Priority Based-Scheduling* berbentuk Grafik Garis

Dari hasil Tabel 6.5 dan Gambar 6.2 di atas maka dapat disimpulkan bahwa task yang memiliki prioritas tertinggi yaitu *Task LDR* akan dijalankan lebih dahulu diikuti dengan *task* yang memiliki prioritas kedua yaitu *Task DHT11*. *Task* yang dijalankan terakhir kali adalah *Task NRF24L01* karena memiliki prioritas terendah. Penjadwalan waktu eksekusi tiap *task* ditentukan dengan *vTaskDelay()*. Penjadwalan waktu eksekusi masing-masing *task* yang telah ditentukan dalam satu periode waktu adalah 3 detik. *Task* yang memiliki prioritas tertinggi akan dijalankan terlebih dahulu kemudian diikuti dengan prioritas yang lebih rendah. Masing-masing *task* telah diberikan waktu eksekusi dalam satu periode waktu adalah 3 detik. Di dalam pengujian ini dapat diketahui ketika waktu eksekusi *task* telah mencapai 3 detik, maka *task DHT11* yang memiliki prioritas lebih rendah akan di *interrupt* oleh *task LDR* yang memiliki prioritas lebih tinggi. Setelah *task LDR* yang memiliki prioritas lebih tinggi selesai dieksekusi, baru melanjutkan kembali eksekusi *task DHT11* yang belum selesai.

BAB 7 PENUTUP

Bab ini membahas tentang kesimpulan dan saran yang diperoleh dari rumusan masalah penelitian dengan melakukan analisis dan pengujian.

7.1 Kesimpulan

Berdasarkan pengujian dan analisa yang dilakukan terhadap tugas akhir ini maka didapatkan kesimpulan sebagai berikut :

1. Implementasi sensor node dapat diimplementasikan pada sistem dengan merakit komponen-komponen yang telah ditentukan seperti Arduino Nano sebagai mikrokontroler, sensor DHT11, sensor LDR dan nRF24L01. Sensor DHT11 dan LDR pada node *client* mampu mengambil data secara simultan lalu data sensor dikirimkan kepada node *base* yang berfungsi sebagai penerima data sensor.
2. Implementasi RTOS ke dalam sistem node menggunakan *library* FreeRTOS yang tersedia untuk mikrokontroler Arduino. *Task-task* yang ada pada node *client* berjumlah 3 yang didefinisikan sebagai *task* LDR, *task* DHT11, serta *task* nRF24L01 yang berfungsi sebagai pengiriman data sensor, sedangkan *task* pada node *base* berjumlah 2 didefinisikan sebagai *task* nRF24L01 yang berfungsi sebagai penerimaan data sensor dan *task* pencetakan data sensor kemudian ditampilkan melalui *serial monitor*. Pengiriman dan penerimaan data sensor pada penelitian ini berjalan secara *real-time* karena node *base* akan menunggu pengiriman data sensor yang dilakukan oleh node *client*. Ketika data sensor belum dikirimkan oleh node *client* maka node *base* tidak akan mencetak data sensor.
3. Implementasi metode Preemptive Priority Based-Scheduling pada sistem node dengan memberikan masing-masing task prioritas yang berbeda dengan tujuan agar sistem dapat diterapkan metode *Preemptive Priority Based-Scheduling*. Prioritas yang tertinggi diberikan pada *task* 1 yang berisi program sensor LDR, prioritas yang kedua diberikan kepada *task* 2 yang berisi program sensor DHT11, dan prioritas yang terendah diberikan kepada *task* 3 yang berisi program pengiriman data sensor menggunakan media komunikasi *wireless* nRF24L01. Pemberian prioritas ini dimaksudkan supaya tersedianya data sensor yang akan dikirimkan melalui nRF24L01 kepada node *base*. Pada implementasi metode ini, *task* 1 berhasil melakukan *interrupt* pada *task* 2 yang sedang berjalan.

7.2 Saran

Berdasarkan pengujian dan Analisa atas penelitian tugas akhir ini maka ada beberapa saran dari penulis yang dapat digunakan untuk melakukan pengembangan sistem ini antara lain :

1. Untuk penerapan RTOS pada sensor node dapat ditambahi lagi jumlah node yang ada supaya data yang dikirim lebih bervariasi.

2. Untuk penentuan `vTaskDelay` dapat dilakukan lebih dinamis lagi sesuai dengan waktu eksekusi tiap *task*.
3. Disarankan diterapkannya *state machine* supaya kinerja dari RTOS lebih efisien.



DAFTAR PUSTAKA

- Arduino CC, 2016. [Online]
Available at: <https://playground.arduino.cc/Main/DHT11Lib>
[Accessed November 2016].
- Arduino CC, 2016. [Online]
Available at: <https://www.arduino.cc/en/Main/ArduinoBoardNano>
[Accessed November 2016].
- Arduino cc, 2016. *Detail of sensor DHT11.* [Online]
Available at: <https://playground.arduino.cc/Main/DHT11Lib>
[Accessed November 2016].
- Arduino Corporation, 2016. *Detail of NRF24L01.* [Online]
Available at:
<https://playground.arduino.cc/InterfacingWithHardware/Nrf24L01>
[Accessed November 2016].
- Cisco, 1984. *cisco corporation.* [Online]
Available at: <https://www.cisco.com/c/en/us/solutions/small-business/resource-center/networking/wireless-network.html>
[Accessed 2018].
- Ltd., Real Time Engineers, 2014. *FreeRTOS Project.* [Online]
Available at: www.freertos.org
[Accessed 14 November 2016].
- Maspermono, 2015. *Fungsi dan prinsip kerja sensor LDR.* [Online]
Available at: <http://belajarelelektronika.net/pengertian-ldr-fungsi-dan-prinsip-kerjanya/>
[Accessed November 2016].
- Pandu, 2012. *Techijau.* [Online] Available at: www.techijau.com [Accessed September 2018].
- Saputro, H. D., 2018. Implementasi Real Time pada Pergerakan Robot Quadruped menggunakan Multisensor dan RTOS. *Journal Pengembangan TIJK*, Volume II, pp. 6868-6875.
- Shobrina, U. J., 2018. Analisis Kinerja Pengiriman Data Modul Transceiver NRF24L01, Xbee dan Wifi ESP8266 Pada Wireless Sensor Network. *Jurnal pengembangan TIJK*, Volume II, pp. 1510-1517.
- Suparlin, A. L., 2018. Implementasi System Real Time untuk Monitoring Pencahaya Suhu dan Kelembaban pada Tanaman Stroberi. *Jurnal pengembangan TIJK*, Volume II, pp. 3278-3286.
- Suprayogi, D. A., 2017. Implementasi Pervasive Service Discovery Protocol pada Rumah Cerdas Berbasis NRF24L01. *Jurnal pengembangan TIJK*, Volume I, pp. 1251-1259.

Wahyudin, I., 2018. Rancang Bangun Sistem Multi-Sensor Untuk Pengukuran Jarak Secara Simultan. *Jurnal Pengembangan TI*, Volume II, p. 8.

